AD_____
(Leave blank)

Award Number: W81XWH-12-C-0154

TITLE: "Enabling Medical Device Interoperability for the Integrated Clinical Environment"

PRINCIPAL INVESTIGATOR: Julian M. Goldman, MD

CONTRACTING ORGANIZATION: Massachusetts General Hospital, Boston, MA 02114

REPORT DATE: August 2013

TYPE OF REPORT: Final Phase I

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT:

√ Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
August 2013	Annual	30 July 2012 – 29 July 2013

4. TITLE AND SUBTITLE
"Enabling Medical Device Interoperability for the Integrated Clinical Environment"

5a. CONTRACT NUMBER
W81XWH-12-C-0154

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Julian M. Goldman, MD
Susan F. Whitehead

email:jmgoldman@partners.org

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Massachusetts General Hospital
Boston, MA 02114-2554

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Medical Research
and Materiel Command
Fort Detrick, Maryland 21702-5012

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This award reflects new and emerging technologies and research, and builds on prior and current MD PnP program work (awards #W81XWH-06-1-0651 and W81XWH-09-1-0705), to develop tools, applications, and sharable databases to advance the state of the art of medical device interoperability and enable a broader community of developers to implement medical device interoperability. This year we identified requirements for an ICE Data Logger and worked with NIST to develop an initial ICE Data Logger research prototype. We built an alpha version of the Clinical Scenario Repository, and started an open-source code-sharing environment on SourceForge where our project code is available for downloading. We implemented CONNECT as part of an ICE system demonstration in the ONC area of the Interoperability Showcase at HIMSS in March 2013. In August 2013 we spent two days at NIH presenting a series of demonstrations of our work for invited representatives from federal agencies; these demonstrations included the initial research prototype Data Logger and Clinical Scenario Repository.

15. SUBJECT TERMS
Medical device, plug-and-play, interoperability, patient safety, health care, standards, data logger, clinical scenario, integrated clinical environment, code-sharing

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			USAMRMC
U	U	U	UU	63	**19b. TELEPHONE NUMBER** *(include area code)*

This page intentionally left blank.

Table of Contents

Page

Annual Report: Enabling Medical Device Interoperability for the Integrated Clinical Environment
Award Number W81XWH-12-C-0154
Principal Investigator: Julian M. Goldman, MD
Period of Performance: 30 July 2012 – 29 July 2013

Introduction

Health Information Technology (HIT) systems should facilitate the collection and point-of-care access to accurate, comprehensive, contextually rich clinical data for all acuity levels of healthcare. Open platforms of plug-and-play medical devices and clinical information systems could enable improved quality and timeliness of data, as well as cost-effective development of innovative third-party medical "apps" for diagnosis, treatment, research, safety and quality improvements, equipment management, and adverse event detection and reporting.

The Medical Device "Plug-and-Play" (MD PnP) Interoperability program was established in 2004 to lead the development and adoption of open standards and related technologies in order to achieve this vision. The MD PnP program is affiliated with Massachusetts General Hospital (MGH), CIMIT (Center for Integration of Medicine and Innovative Technology), and Partners HealthCare System, Inc., with additional support from TATRC (U.S. Army Telemedicine & Advanced Technology Research Center). Having evolved from the Operating Room of the Future program at MGH, the MD PnP program remains clinically grounded. We have taken a multi-faceted approach to address key barriers to achieving interoperability, including the development and support of suitable open standards (e.g. ASTM F2761-09, Integrated Clinical Environment "ICE"); the elicitation, collection and modeling of clinical use cases and system engineering requirements for an open architecture instantiation of ICE as a platform and "ecosystem"; alignment of clinical organizational, manufacturer, and FDA regulatory expectations; and implementation of prototype use cases in an open "sandbox" environment.

The MD PnP program has built a geographically dispersed, interdisciplinary, multi-institutional team to develop and implement a strategy to address historical barriers and accelerate the achievement of device interoperability through collaboration. Since the program's inception, more than 850 clinical and engineering experts, and representatives of more than 120 companies and institutions have participated in our plenary workshops / conferences, working group meetings, and focus groups to contribute to ongoing program activities that helped shape the common goals.

TATRC support for MD PnP program development has enabled significant progress towards the goal of achieving medical device interoperability. TATRC's funding has leveraged additional synergistic project-specific funding from CIMIT, NSF, NIST, and NIH, but it is TATRC funding that has uniquely made possible our program's enabling efforts that are moving medical device interoperability and patient safety forward along parallel pathways of requirements, standards, platform development, and regulatory approach. A major outcome of TATRC funding has been enabling our team to form and grow a diverse community of involved and committed collaborators and stakeholders. That context has enabled the work of this award to focus on the key elements of ICE system data logging, web-based reporting of clinical scenarios to spur innovative integrated solutions, open source code dissemination, and ICE external interface data transfer to other health IT systems.

Body of Report

The goal of the Medical Device "Plug-and-Play" (MD PnP) Interoperability program is to accelerate medical device interoperability to enable the creation of complete and accurate electronic health records and the cost-effective development of innovative third-party medical "apps" for diagnosis, treatment, research, safety and quality improvements, equipment management, and adverse event detection and reporting when integrating networked medical devices for clinical care. This award reflects new and emerging technologies and research, and builds on prior and current MD PnP program work (awards #W81XWH-06-1-0651 and W81XWH-09-1-0705), to develop tools, applications, and sharable databases to advance the state of the art of medical device interoperability and enable a broader community of developers to implement medical device interoperability. Moreover, the Clinical Data Repository can be used by patient safety organizations and may generate data to support healthcare policy changes.

For the period of this award, we proposed the following aims:

Aim 1: ICE Data Logger
Develop a software research prototype of the Data Logger component conforming to the ICE standard (ASTM F2761). Data logging is necessary to address regulatory and liability concerns regarding networked medical device systems, and will improve the forensic analysis of clinical adverse events and near misses.

- Base the prototype on requirements identified through the NIH Quantum project
- Develop an event recording and playback capability that demonstrates the potential for forensic analysis of activity in networked medical device systems, as well as improved adverse event analysis (useful for hospitals, FDA, manufacturers)
- Perform an assessment of the FDA Unique Device Identifier, and use it (if available during this period of performance) to identify devices in the data log
- Validate the clinical usefulness of the Data Logger by analyzing simulated adverse events
- Publicly disseminate research results

Aim 2: Web-Based Clinical Scenario Repository
Develop a sharable repository of clinical scenarios that could be improved through better medical device and health IT integration. The scenario repository will provide use cases to inform design of the Data Logger, and can eventually be used by researchers, standards developers, regulators, and manufacturers to create innovative solutions for many intractable clinical problems.

- Provide a web portal to allow clinicians, clinical engineers, and other users to enter, revise, and annotate clinical scenarios
- Design database back-end and administrative system to organize users and permissions
- Use feedback from the FDA, NIST, VA, industry, and other potential users to enhance usability
- Make repository available to broader MD PnP community for research, standards development, etc.
- Publicly disseminate details of repository

Aim 3: Open Source Code Dissemination
Disseminate open-source code developed by the MD PnP program and collaborators, including the prototype Data Logger, in order to facilitate further development by others.

- Determine appropriate venues, tools, and processes for releasing code
- Help interested external parties to obtain code and documentation
- Manage the integration of external code that is received into official releases

Aim 4: ICE External Interface Data Transfer
Define and document external interfaces to bi-directionally transfer medical device and patient contextual data between the integrated clinical environment and external systems of national interest. Demonstrate the interface to/from one or more of these systems (depending on which are ready and accessible):

- The VHA Open VistA EMR
- The Nationwide Health Information Network (NwHIN) and the local Massachusetts health information exchange
- The evolving multi-agency Health IT Innovative Development Environments (HITIDE)
- The ONC Pan-SHARP project demonstration
- Publicly disseminate research results

Research Accomplishments

ICE Data Logger, Aim 1: Develop a software research prototype of the Data Logger component conforming to the ICE standard (ASTM F2761). Data logging is necessary to address regulatory and liability concerns regarding networked medical device systems, and will improve the forensic analysis of clinical adverse events and near misses.

The MD PnP team compiled an initial set of needed attributes and technical requirements for the ICE Data Logger that specify what data will be recorded, the format of the data, the timestamping, cryptographic signature, and sequencing of data, and other technical details. These requirements also cover data playback, particularly where features of the Data Logger will influence what playback capabilities are possible.

Requirements were based upon:

- Content from the Integrated Clinical Environment (ICE) standard (ASTM F2761-09)
- Experience to date in the MD PnP interoperability lab
- Work with our collaborators on the Quantum Medical Device Interoperability (QMDI) project funded by NIH
- Early data logger concept and an MD PnP paper presented at the International Conference on Biomedical Ontologies
- Clinician Interviews
- An FDA conceptual design for a stand-alone device data log

We planned to build a Data Logger implementation following these requirements, with the expectation that building it would reveal necessary refinements to the requirements, resulting in future iterations of the requirements document. We have updated and maintained these requirements to reflect lessons learned during development, as well as changes to the QMDI requirements that specify the system in which the Data Logger will operate.

Several months into our data logger work, a research group at NIST approached us wanting to collaborate on this project, using their own internal funding. Starting with documentation, requirements, and guidance from our team, NIST then surveyed relevant data logger work in avionics, automotive, and other domains to identify additional requirements. NIST compiled this set of data logger requirements, and has written a technical white paper about the different levels or modes of logging that an ICE Data Logger will need to support. We plan to write a paper, in collaboration with both NIST and FDA, based on this work that will compare our ICE Data Logger to data loggers in other domains, in order to highlight the specific needs that differentiate clinical data logging from aerospace or automotive data logging.

This detailed comparison with other data loggers, such as aircraft flight data recorders and automotive loggers, has enabled us to work with NIST to leverage their unique engineering expertise to help build a consensus set of requirements to feed back to the QMDI project and the broader community and to use to develop an ICE Data Logger standard.

We also worked with NIST to plan their implementation of a research data logger prototype – implementing many of the identified requirements, and intended to demonstrate some of the challenges and our approach to logging and playback for ICE – on their in-house Data Flow System. This prototype implementation was demonstrated as part of a set of MD PnP and collaborator demonstrations held at NIH on August 21-22 2013, attended by TATRC and many other federal agencies (including FDA, NIST, NSF, NIH, ONC, and others). We will share the documentation and code for this data logger prototype through our public SourceForge site.

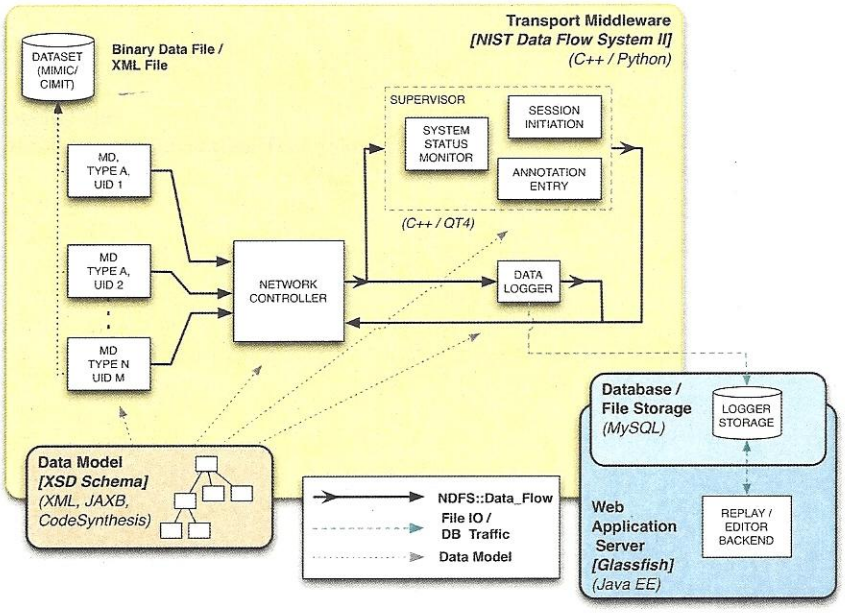


Figure 1: The ICE Data Logger & Playback Demonstration Architecture

This logger and playback prototype is based on our requirements documents, but built using NIST's Data Flow middleware and data collected from medical devices in our MD PnP lab. NIH funding this past year enabled us to purchase new medical equipment for the lab, including two systems from third-party medical device integrators. The Bedmaster system allows for collecting data from a GE medical network, and is cleared by the FDA for this use. Cardio-Pulmonary Corporation (CPC) makes a system called Bernoulli that is similarly approved by FDA for the purpose of collecting data from a variety of devices. Because these systems are cleared by FDA for these uses, it is possible for hospitals and researchers to use them in clinical settings.

The Bedmaster and CPC integration systems impose their own restrictions on what data is available and on the data's timeliness. To create data for NIST to use in developing their data logger prototype, one of our engineers configured the Bedmaster system and the CPC system to log the data and then export it in a useful format. This required setting up the devices and patient simulator, starting data collection, changing settings on the simulator to create a particular clinical scenario, exporting the logged data, and then uploading the data files to SourceForge, where the data were available for NIST and are publicly available for anyone else who might find such data useful for research. During some of the sample runs, we also made video recordings of the patient simulator and devices, so that the data logger playback application could display synchronized video.

The synchronized video is important for revealing clinical context. For instance, part of the data logger demo at NIH in August included showing a scenario where the patient received an overdose from a PCA pump. The device data shows the patient's physiologic response, the log from a PCA pump would show that the dose request button was pressed, but only the video can reveal that the button was pressed by someone other than the patient (an example of PCA-by-proxy). Thus, the root cause of the patient's overdose can only be found by integrating a video record with the device data.

We began looking at Data Logger performance testing in the context of the collaborative NIST prototype implementation. This initial prototype data logger implementation was built on NIST's Data Flow System, which is designed to handle extremely large amounts of data. Our simulators were not able to generate enough traffic to stress-test the NIST middleware, so it is more than sufficient for our applications. However, if we planned to stay with the Data Flow System longer term, we would need to create medical data simulators that could send data much faster in order to test the limits of the system. We plan instead for the next Data Logger implementation to move away from NIST's in-house framework and onto the DDS middleware we are using for the SourceForge implementation, and to use our ICE Equipment Interfaces directly rather than using data from the third-party integrators.

DDS is an open standard from OMG. The DDS implementation we are using is built by the RTI company, and is used extensively in DoD applications like ship "command and control" networks and drone avionics. These applications require high performance and reliability, and RTI has extensively tested the performance of the middleware. We have worked with RTI to make their tools freely available to our community through an ICE Community License. This makes all of their code and tools available at no cost for research and prototyping. Use in commercial products would, of course, require licensing from RTI, or using a different implementation of the open DDS standard.

Once we move the prototype data logger to our DDS-based open source implementation, we will perform extensive end-to-end performance testing to ensure that the entire system will be able to handle large amounts of data. It is important for us to quantify the maximum data throughput of the system. This will ultimately be limited by the speed of the data storage system, and these results will allow us to specify the storage requirements for individual ICE data loggers and also for an architecture where all ICE data is backed up in a central data store at the Healthcare Delivery Organization.

One particularly rewarding outcome from the NIH demos was learning that the NIST development team is quite excited about helping us to move the data logger to our framework, and in improving the shared infrastructure in our open-source implementation. It was also clear from the audience at the NIH that there is considerable and widespread interest in this work.

Web-Based Clinical Scenario Repository, Aim 2: Develop a sharable repository of clinical scenarios that could be improved through better medical device and health IT integration. The scenario repository will provide use cases to inform design of the Data Logger, and can eventually be used by researchers, standards developers, regulators, and manufacturers to create innovative solutions for many intractable clinical problems.

Objectives for this first year included building and testing a robust preliminary web-based prototype of the Clinical Scenario Repository, leveraging earlier work done under TATRC award W81XWH-09-1-0705. We invested substantial effort in a careful design and implementation that facilitates both administration and general usability of the Clinical Scenario Repository.
The clinical scenario repository web application has been totally rebuilt from the original prototype, with numerous additions to functionality that have been more efficiently implemented

using newer frameworks. We have used a more common toolkit for building the site that gives it modern features, e.g. data is saved automatically as a draft in progress while the user is working, with the option to manually “Save for later”, and at the end of data entry, the user is given the option to “Submit for approval”.

An alpha version of the prototype Clinical Scenario Repository has been completed, which was demonstrated in August to TATRC and other federal agencies (see below). The Clinical Scenario Repository has user features to create new scenarios and search the existing database of scenarios, and system administrator features to review, approve, and manage scenarios. The current version of the application is hosted on the Google Application Engine, which provides an easy and reliable way of managing the user log-in process, email communications, and data storage. Complete features include the user registration and log-in process, as well as the persistence of administrative user information, the data from the scenario description, and other related data (such as keywords to tag the scenarios for search/indexation purposes).

Despite some challenges to adopt the Google Application Engine’s non-traditional approach to the development of some of these features, we think that its capabilities for scaling and balancing traffic requirements, reliability of servers, and features for data management will prove invaluable in the near future, as we research deeper into managing the information contained in the different scenarios in the repository. Using the Google Web Toolkit to develop the front-end part of the application has already proven a wise choice, since it enabled our developers to catch up easily with the web development technologies necessary to implement the browser side of the web portal.

Our implementation includes a database schema that is a *superset* of the data specified in the clinical scenario template of Annex B of ASTM standard F2761-09 for the Integrated Clinical Environment (ICE). We normalized the schema to make it robust enough for the higher traffic we anticipate on a generally available web site. We formally defined the *state model* for a scenario (e.g. In Progress, Pending Approval, Approved, etc.), which has become a challenging task because the rules for transitioning from one state to another – or even the number of states – might change as we develop new features, receive feedback from our collaborators, and consider different behaviors in the users’ interaction with the application.

Basic user roles have been defined (unregistered visitor, registered collaborator, and system administrator), and a coherent set of functionalities and privileges have been granted to each role. For example, system administrators can view all scenarios submitted, registered collaborators are able to view all approved scenarios as well as scenarios they have themselves entered, regardless of status, and unregistered visitors can view only scenarios that have been approved and are part of the viewable database – they cannot enter a scenario unless they register. We added the “unregistered visitor” role as a way to show some utility to a new visitor and provide them a motivation to register with our site.

Basic and advanced (specific) searches of approved scenarios are available to look for scenarios containing certain keywords or meeting certain conditions (such as type of clinician involved, severity of a hazard, etc.). Additional features allow users to manage their own contributions, and allow system administrators to detect new submissions pending review and to take actions such as approval or request of further clarification.

Major features:

- **Registration:** The Google Application Engine provided a registration system for users using Google IDs. This relieved us from implementing our own registration system and requesting, encrypting and securely managing and maintaining usernames, passwords

and other personal information from users; this allowed us to focus on the features at the heart of the repository. In the near future this registration process will be extended to include “OpenID federated login” and other existing providers for Secure-Socket-Layer authentication and registration. The personal information shared in the registration process is kept private and is not shared with other users.

- **Scenario Entry:** For scenario entry our design follows a tabbed “breadcrumb” approach, allowing the user to move easily between sections of the scenario entry process without enforcing a strict path through those sections. This will allow *Registered Users* to immediately enter the information they have readily available, and to easily return later to complete other sections. At the top of each text entry box, we include pop-up menus to provide an "example scenario" to show what to fill in. The clear explanation of fields will help users to enter more useful data, and we will continue to explore additional ways to provide contextual assistance.
- **Search:** Our search functionality follows a “keyword” approach, offering the user the ability to search all data fields for keywords of interest.
- **Approval Workflow:** Our Repository Administrator will be able to view new pending scenarios submitted by *Registered Users*, and will review and approve scenarios before they become part of the public repository. In anticipation that content clarification will be needed for many submissions, we have facilitated communication between submitters and approvers using an email feature.

  Traditionally web apps have followed a simple “form submission” model where a user fills out numerous fields and clicks *Submit*. We are using a more modern approach (AJAX mechanism) that allows us to save a user’s progress while they are working in order to ensure no data is lost. This introduces new challenges – for instance, we must store and manage all of a user’s current draft work. We also must make decisions about how often and with what granularity to send data back to the server.

  For the storage of data, using Google App Engine has created new decisions for us to make. One option is to use a more traditional Relational Database Management System (RDBMS) hosted either in the cloud or on our own servers. A second option is to use the Google High Replication Datastore, which is a “big data” technology that scales far better than an RDBMS but creates other issues. At this point in development, we are staying within the confines of an abstraction layer (“Java Data Objects”) that supports either storage subsystem. As we avail ourselves of more advanced storage features, we may need to make a decision about which technology to utilize.

The prototype Clinical Scenario Repository is based on the template designed by the MD PnP Program for describing and documenting information related to clinical scenarios and use cases that could benefit from medical device interoperability. The application requests information from the user following an easy and descriptive approach utilizing a series of tabs:

- **Scenario Description:** is where the user can describe in detail the adverse event or clinical challenge – the *current state*. They can also describe the enhancement in safety that can be accomplished by an integrated solution in a *proposed state*.
- **Hazards:** is used to describe the factors contributing to the risk represented by the scenario, including their level of severity and the expectation of occurrence.
- **Environments:** is used to capture the clinicians involved in the scenario (e.g. nurse, anesthesiologist, surgeon, etc.) and the environment where it took place (e.g. operating room, hospital ward, ambulance, etc.).
- **Equipment:** is used to describe the medical devices or sensors that play an important role in the scenario.

- **Proposed Solution:** allows for a more extensive description of an ideal state or workflow, and how it might affect or change the practice environment.
- **Benefits and Risks:** is used to gather information about the obstacles eliminated by the new process, as well as any new risks that might be introduced by the proposed solution, so these can be mitigated in advance.
- **Feedback:** available only to administrators reviewing a submitted scenario, this tab is used to approve the scenario (granting any registered user permission to see it) or to request clarification from the scenario submitter via email.
- We have also foreseen the potential for a **References** tab, where users could add relevant references or publications related to the scenario described. An earlier version of the repository included this feature, but we recognized the need for further consideration about the kinds of web links, images, and/or documents users should be able to include as references – we will clarify and resolve this question before introducing this feature in the current implementation.

Users can save the scenario information at any time, allowing them to enter the information available at the moment and to revisit these tabs at another time to complete or update the information. This approach relieves the user of being forced through a multitude of input fields and constrained data input workflow processes. We received positive feedback on the intuitive navigation from NIH demo attendees.

The culmination of our first year's work was the opportunity to show the alpha version of the prototype Clinical Scenario Repository when we presented a series of demonstrations of our work at NIH on August 21-22 2013 for invited representatives from federal agencies. Over 60 visitors from DoD, FDA, NIST, NIH, and other federal agencies attended, and we received positive feedback, encouraging us to develop additional features, e.g. advanced search capabilities that might include an ontology of terms and use of natural language processing of submitted text to auto-create keyword tags. A set of slides showing screen shots of the alpha Repository is included as Appendix 3.

We are testing a new capability (currently available only to system administrators) to create keywords to associate with specific scenarios, thus making indexing, searching and information extraction from the repository easier, quicker and more meaningful. We expect that ongoing feedback about the scenario search results will provide insights into how these tags should be created, reviewed, managed, and associated correctly to the scenarios.

During the next quarter we will continue to incorporate feedback on the alpha prototype, and we plan to have a beta version that will be shared with external collaborators, TATRC, and representatives of other federal agencies for feedback. This will help to identify usability issues, documentation needed, and further functionality to be developed, e.g. tools for data-mining of the information contained in the repository.

Open Source Code Dissemination, Aim 3: Disseminate open-source code developed by the MD PnP program and collaborators, including the prototype Data Logger, in order to facilitate further development by others.

We began working with Open Health Tools (OHT) in March 2012 to consider a process for sharing code and other tools. This relationship has informed our thinking about the challenges of sharing code and the possible approaches. In addition, our NIH QMDI sponsors have strongly and consistently encouraged us to share code and other artifacts from that work, but the TATRC award has enabled us to do the necessary research and organization to develop a plan and an open-source approach for doing so.

We began in September 2012 posting several projects on GitHub, a popular open-source project hosting platform. However, we quickly identified limitations in tracking page views and downloads of source code. For this reason, we started hosting projects in March on SourceForge, which supports more metrics: http://sourceforge.net/projects/mdpnp/. Unlike GitHub, SourceForge allows us to easily share artifacts that are not source code. For instance, we have been obtaining ECG and pulse oximeter data from a GE Central Station and patient monitor, and have posted this data on SourceForge for use by other researchers.

Since March we have added a diverse set of software to our code repository on SourceForge. This site has become the focus of all development activity for MD PnP for this TATRC award, our NIH U01, and other projects. Our repository includes software components for interfacing with devices in our interoperability lab, as well as the speculative software we have built to connect those devices and implement demonstration applications. By making all of our work available at an early phase of development, we hope to involve the broader research community as much as possible. We have recorded hundreds of downloads from dozens of countries since we launched the repository (see Figure 2 for activity over this time period). This site has already become a key point of synchronization with our collaborators, and we have had some initial success in getting our collaborators to also commit changes back into the system.

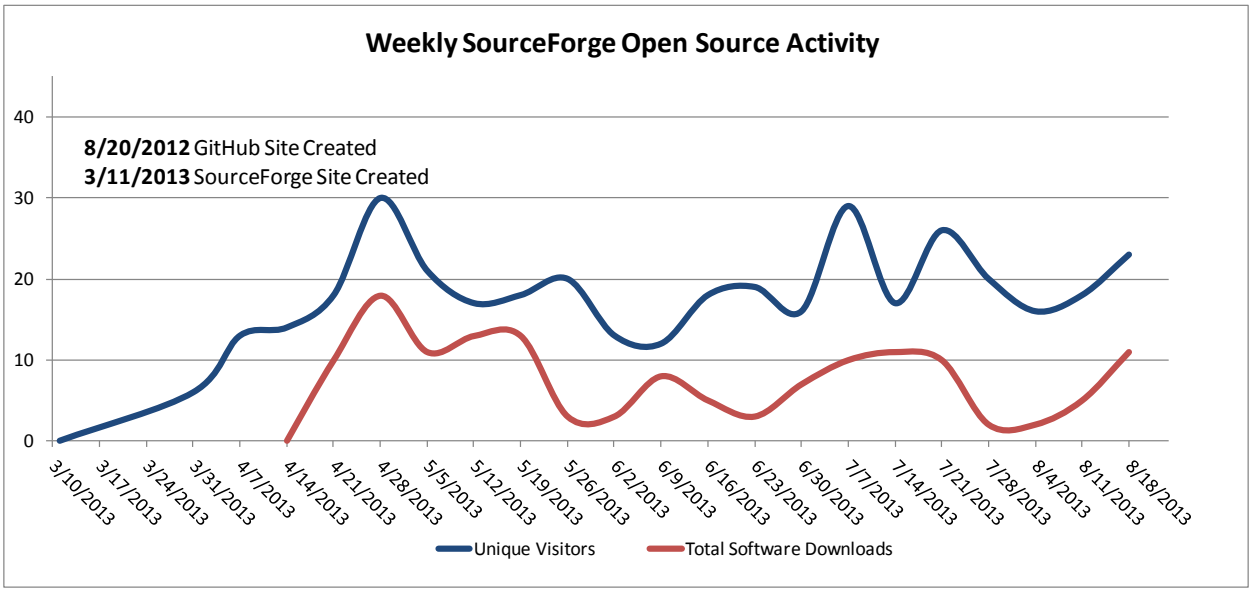


Figure 2. Weekly SourceForge Access Activity March – August 2013

The software has been organized into a coherent build, and a continuous integration server has been set up. A coherent build system makes it easier for community members to modify the code because it automatically creates an environment on their computer amenable to building the software (gathering third party libraries, configuring the compiler, etc). Continuous integration enables us to monitor changes made to the code repository and it reports in real time on any changes to the code that prevent its building successfully or any failed unit tests. We have been exercising these processes among our own team as preparation for involvement of the broader community. Figure 3 shows the types of code changes being made in the SourceForge repository.

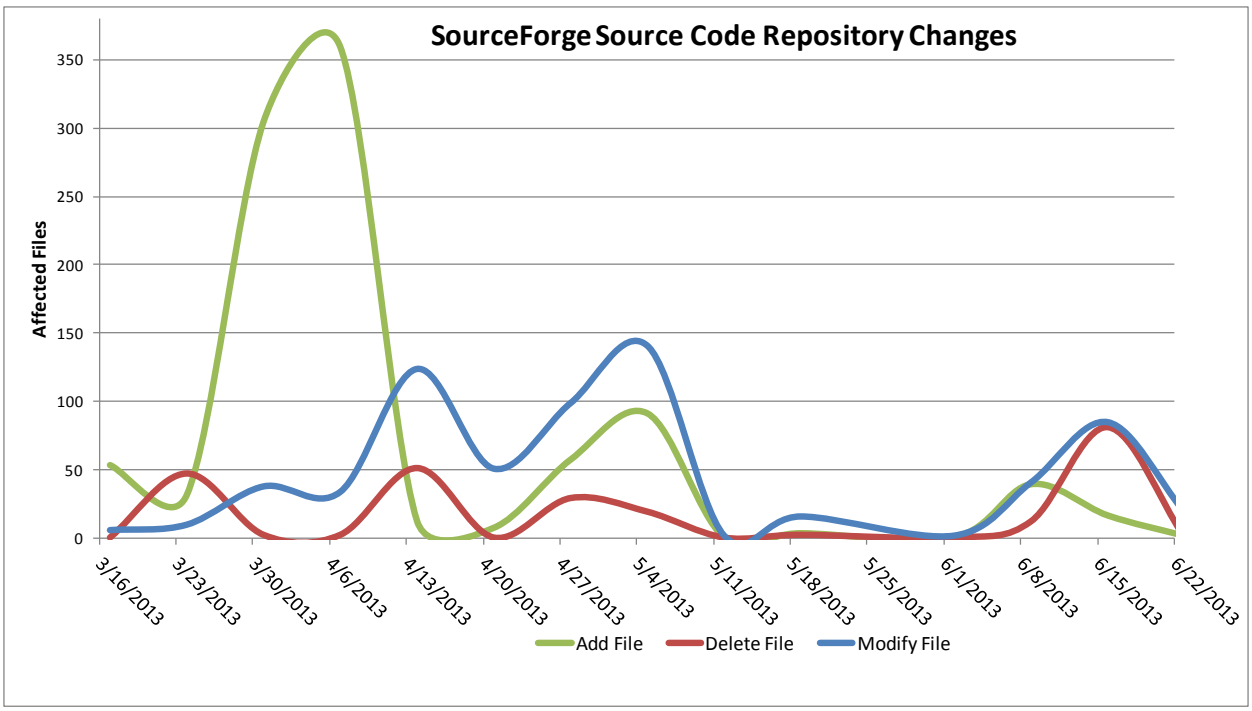


Figure 3. Weekly SourceForge Code Changes March – August 2013

We have not yet focused on publicizing our repository, as we've been adding material and essentially beta-testing it with our collaborators. Even without publicity, we have people finding our code and downloading it. We have not yet been proactively building the community to gather feedback from these "users" or to ask them to become contributors. Both of these activities are goals for the coming year.

We are also continuing our association with Open Health Tools, and we will be hosting their board meeting in September 2013.

ICE External Interface Data Transfer, Aim 4: Define and document external interfaces to bi-directionally transfer medical device and patient contextual data between the integrated clinical environment and external systems of national interest. Demonstrate the interface to/from one or more of these systems (depending on which are ready and accessible).

To achieve this aim, we built on learnings from a CIMIT-sponsored project on Veterans Healthcare Data Exchange, which involved connectivity and exchange of data between the Partners HealthCare electronic health record and both the VistA and AHLTA systems. While limited in scope, by design, that project provided a good foundation for the bi-directional transfer work on this TATRC project, including the establishment of good relationships with contacts at both TATRC and the VA.

We built on this earlier CONNECT and DIRECT work to develop a technology demonstration of our use of CONNECT in an ICE system demonstration at HIMSS13 (Healthcare Information & Management Systems Society annual conference) on March 4-7 2013. Our demo was selected by the Office of the National Coordinator for Health IT to be part of the ONC's demonstration area in the Interoperability Showcase. The MD PnP team collaborated with DocBox Inc. and Kansas State University to produce a demo on "Transferring a Patient's Device Settings between Care Environments," which showed the importance of device data as part of national interoperability efforts.

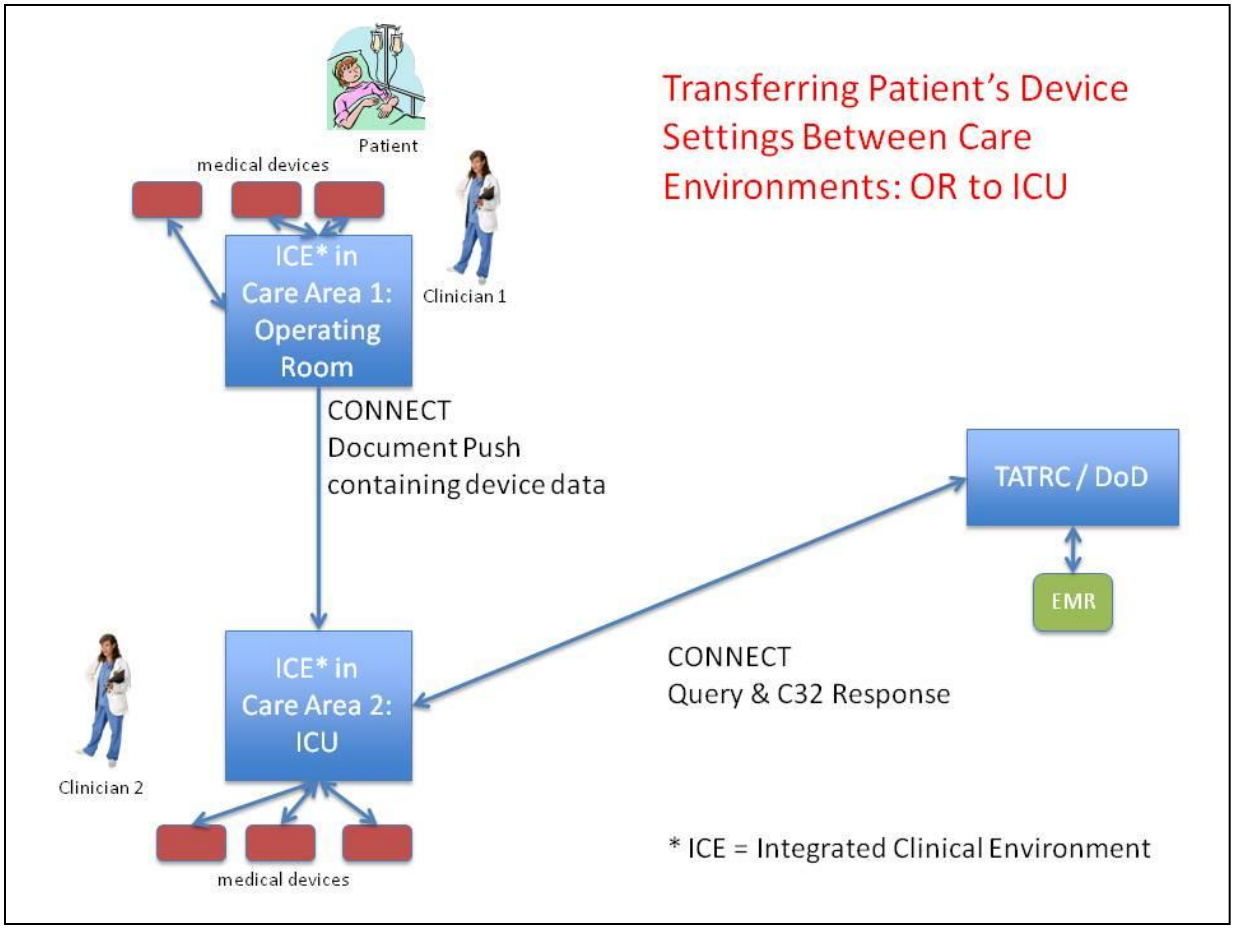


Figure 4. HIMSS13 Demo on Transferring Device Settings Between Care Environments

The demonstration (see Figure 4) showed connectivity between two ICE systems (standards-based Integrated Clinical Environments) in the OR and ICU, and use of the NwHIN to automatically return current device data in response to a clinician query. The demo showed reading and changing of device settings between the OR and ICU, external query via CONNECT to the TATRC test EMR with a return of allergy information, coordination between multiple apps, and coordination via CONNECT between a commercial ICE implementation (the TATRC-sponsored system from DocBox) and a research / rapid prototyping ICE implementation using the open-source Medical Device Coordination Framework (MDCF) provided by collaborators at Kansas State University.

The HIMSS demo was visited by over 300 HIMSS attendees, and was one of the few ONC demos visited by the National Coordinator for Health IT, Farzad Mostashari – he said it was the "most exciting" demo in the ONC area.

This aim is complete insofar as we have been able to connect with the federal systems that are ready and accessible to us: AHLTA and VistA, and we have published our CONNECT code on SourceForge. We will continue to seek opportunities for further connectivity if HITIDE becomes available, and with the Massachusetts HIE. The Pan-SHARP program is no longer active.

Milestones:

Milestone	Aim	Qtr Due	Status
Data Logger requirements	1	Q1 Rpt	Completed
Data Logger software architecture description	1	Q3 Rpt	Completed
Results of Data Logger performance testing	1	Annual Rpt	Completed
Scenario Repository web portal example screen captures	2	Q2 Rpt	Completed
Repository interface for alpha testing	2	Q3 Rpt	Completed
Repository software architecture description	2	Annual Rpt	Completed
Code release plan	3	Q3 Rpt	Completed
Results of Data Exchange performance testing	4	Annual Rpt	Completed

- Data Logger Requirements are included as Appendix 1 – these will be updated in the next quarter.
- Data Logger software architecture description – Appendix 2 is the preliminary description of the Data Logger software architecture, as presented in a summary of the joint NIST-MD PnP work – this will be updated in the next quarter.
- Data Logger performance testing is described within the Aim 1 section of this report.
- Slides including screen shots from the alpha version of the Clinical Scenario Repository are included as Appendix 3.
- Repository interface for alpha testing – the source code for the current version is available at http://sourceforge.net/p/mdpnp/code/ci/master/tree/clinical-scenarios/.
- Repository software architecture description is within the Aim 2 section of this report.
- Code release plan – Appendix 4 provides the code release plan, based on work to date.
- Results of the Data Exchange work are part of the HIMSS13 report in Appendix 5.

Synergistic Activities. The activities under this award have enabled the PI and the MD PnP program to remain actively involved with national health IT developments to support inclusion of medical device interoperability on the agenda.

The MD PnP program has continued to work with the FDA NIST, NSF, and the Office of the National Coordinator for Health IT. Recognition of the critical role of device interoperability in the national health IT agenda has increased greatly, as evidenced by the following activities:

- Dr. Goldman has served as invited co-chair of the Regulations Subcommittee of the Food and Drug Administration Safety Innovation Act (FDASIA) Workgroup of the Health IT Policy Committee. In the Subcommittee's final recommendations, the importance of healthcare data logging was cited.

- Our work under this award, as well as our larger body of MD PnP program work, has been foundational to the new AAMI-UL 2800 device safety certification standard and to the new AAMI PCA Safety standard, both of which are under development with participation from our team.

- The Data Logger work under this award is forming the basis of a new ICE Data Logger standard.

- During the past year Dr. Goldman continued to participate in meetings with DoD regarding procurement of medical devices – one of the key requirements is for devices in future to communicate the data needed for interoperability.

- Dr. Goldman continues as part of a group convened by the Brookings Institution to discuss capturing unique device identifiers (UDIs) in administrative health care claims. As part of the UDI Implementation Work Group, we plan to implement and test a UDI for ICE and to provide our results to the FDA, as soon as FDA issues their final rule on UDI. At the August demonstrations at NIH, we showed our early implementation of UDI to FDA personnel who are overseeing UDI and confirmed with them our plans to collaborate.

Key Research Accomplishments

- **Initial prototype ICE Data Logger.** After our initial work on identifying requirements for an ICE Data Logger, we were approached by NIST to collaborate. NIST was interested in the broader applicability of the ICE Data Logger concept and was in a unique position to review design and data aspects of data loggers used in other domains such as avionics and automotive. This collaborative work resulted in an initial prototype ICE Data Logger that was demonstrated to multiple federal agencies in August 2013.

- **Initial prototype Clinical Scenario Repository.** We built an alpha version of the Clinical Scenario Repository that was also demonstrated to multiple federal agencies in August 2013. This will enable a shorter period of alpha testing, to be followed by beta-testing that will involve a broader community of collaborators.

- **SourceForge Code-Sharing Repository**. We started an open-source code-sharing environment on SourceForge in March 2013 where our project code is available for downloading – to date We have recorded hundreds of downloads from dozens of countries.

- **HIMSS13 Demonstration.** We implemented CONNECT as part of an ICE system demonstration in the ONC demonstration area in the Interoperability Showcase at HIMSS13 (Healthcare Information & Management Systems Society annual conference) in March 2013. The HIMSS demo was visited by over 300 HIMSS attendees, and was visited and applauded by the National Coordinator for Health IT, Farzad Mostashari.

- **Demonstrations for Federal Agencies.** In August 2013 we spent two days at NIH presenting a series of demonstrations of our work for invited representatives from federal agencies. These demonstrations included the initial prototype Data Logger and Clinical Scenario Repository. Over 60 visitors from DoD, FDA, NIST, NIH, and other federal agencies attended, and we received very positive feedback.

In addition to the specific achievements above, the MD PnP program has continued to gain increasing traction through our collaborative relationships. The web of connections among people in our community of interest continues to generate new connections to supportive individuals in government agencies, healthcare institutions, and other organizations who are helping to further the aims of the program.

Reportable Outcomes

150+ Meetings:

- August 2012 – July 2013 – weekly teleconference calls of the Medical Device Interoperability Safety (MDIS) working group
- August 2012 – March 2013 – biweekly Pan-SHARP teleconference calls
- August 2012 – July 2013 – 16 teleconference calls for the FDA MDICC activity
- August 2012 – July 2013 – 22 FCC mHealth Task Force teleconferences
- August 2012 – July 2013 – 5 AAMI HTSI Alarm Systems Steering Committee teleconferences
- August 2012 – January 2013 – monthly MDISS: HDO teleconferences
- August 9-10 2012 – UL2800 Standard Meeting, Washington, DC
- August 13 2012 - Deloitte Information Security Program Review (teleconference)
- September 14 2012 – Open Health Tools (OHT) Meeting, Baltimore, MD
- September 24 2012 – FCC mHealth Task Force meeting, Washington, DC
- October 1 2012 – FDA MDICC meeting, Washington, DC
- October 15 2012 – Brookings Institute meeting on Unique Device Identifiers (UDI), Washington, DC
- October 16 2012 – Meeting with DoD acquisition personnel, Washington, DC
- October 31 2012 – MD PnP Lab Open House technology demonstrations, Cambridge, MA
- November 2012 – April 2013 – monthly teleconferences of the UDI Implementation Work Group
- December 3-5 2012 – mHealth Summit, Washington, DC
- December 6-7 2012 – AAMI HTSI Alarms Steering Committee meeting, Daytona, FL, via teleconference
- December 10-12 2012 – hosted ISO TC121 Sc2 standards meeting, Cambridge, MA
- December 13 2012 – FDA UDI meeting at Brookings Institute, Washington, DC
- December 14 2012 – IOM meeting on systems engineering in health, Washington, DC
- March 1 2013 – Meeting with UL, Chicago, IL
- March 8 2013 – Open Health Tools Board of Directors meeting, New Orleans, LA
- March 18 2013 – UDI Workshop at Brookings Institute, Washington, DC
- April – May 2013 – weekly teleconference calls for AAMI / UL 2800 brainstorming
- April 26 2013 – FCC Consumer Advisory Committee meeting, Washington, DC
- May – July 2013 – 17 teleconference calls of the FDASIA Workgroup of the HIT Policy Committee
- May 30-21 2013 – Meeting of the FDASIA Workgroup of the HIT Policy Committee, Washington, DC
- June 3 2013 – AAMI / UL 2800 meeting, Long Beach, CA
- June 4 2013 – AAMI HTSI Alarms Steering Committee meeting, Long Beach, CA

22 Presentations on Medical Device Interoperability Topics:

Dr. Goldman delivered invited presentations on topics related to medical device interoperability for improving patient safety and healthcare efficiency to the following groups during the past year:

- September 11 2012 at MDEpiNet (Medical Device Epidemiology Network) Annual Meeting at FDA, Washington, DC

- October 2-3 2012 – Lectures and panel presentation at FDA AAMI Interoperability Summit, Washington, DC
- October 4 2012 – Panel presentation at NSF CPS PI Meeting, Washington, DC
- October 15 2012 – Panel moderator at American Society of Anesthesiologists (ASA) Annual Meeting, Washington, DC
- October 25 2012 – Presentation at NSF Time Workshop, Baltimore, MD
- November 2 2012 – Keynote and closing panel at Medical Device Connectivity Conference, Boston, MA
- November 4 2012 – Presentation on Pan-SHARP at AMIA Conference, Chicago, IL
- November 5 2012 – Invited lecture at University of Illinois at Urbana-Champaign, Urbana, IL
- November 29 2012 – Panel at Wireless Connectivity in Medical Devices Conference, Boston, MA
- December 3 2012 – Panel moderator at FCC mHealth Summit, Washington, DC
- January 10 2013 – Panel at Society for Technology in Anesthesia Annual Meeting, Phoenix, AZ
- February 16 2013 – Panel at Advancing Science, Serving Society Annual Meeting, Boston, MA
- March 4-7 2013 – Lecture and Technology Demonstrations at HIMSS Conference, New Orleans, LA
- March 4 2013 – Keynote at IBM systems engineering symposium, Waltham, MA
- April 8 2013 – Lecture at CPS Week, Philadelphia, PA
- May 20 2013 – Grand Rounds lecture on interoperability at Tufts Medical Center, Boston MA
- May 23 2013 – Grand rounds lecture on interoperability at Geisinger Health System, Danville, PA
- July 3 2013 – Lecture at meeting of Food and Drug Administration Safety Innovation Act (FDASIA) Regulations Subgroup, Washington, DC

Web Site:

- www.mdpnp.org is maintained as a major communication vehicle for the MD PnP program and is updated frequently. The website provides access to the ICE standard, MD FIRE contracting language, publications, posters, links to streaming video of talks from plenary meetings, and downloads of sharable documents and code via our SourceForge public project at http://www.mdpnp.org/Download_Files.html.

  On the website we advertise General Membership in the MD PnP community, offering updates via our occasional eNewsletter, access to documentation, software, and educational materials, and an invitation to the RTI Infrastructure Community for Implementation of DDS. We currently have 105 members, and the website receives about 1,000 visits per week.

Manuscripts/Publications:

- Arney D, Goldman JM, Bhargav-Spantzel A, Basu A, Taborn M, Pappas G, Robkin M. Simulation of Medical Device Network Performance and Requirements for an Integrated Clinical Environment. Biomed Instrum Technol. 2012 Jul-Aug;46(4):308-15. doi: 10.2345/0899-8205-46.4.308.

This is a report on our work with Intel on network and computer infrastructure design and operations to support interoperability.

Funding Applications Facilitated by this BAA to Date (total costs shown):

- None at this time

Conclusions

As with prior TATRC BAA support, this award has supported activities that are impacting the national healthcare scene. There is an increasing interest nationally in the concept of healthcare data logging and its potential importance both in enabling better forensic analysis of adverse events and in facilitating the safe integration of multi-vendor medical device systems. Our Clinical Scenario Repository will enable foundational work for what could become a new Health IT Safety Board by uncovering broad healthcare problems requiring the alignment of diverse national stakeholders to address those problems. The concept of the necessity of collecting that data has been recognized by FDASIA. When Jacob Reider, the ONC Chief Medical Officer, saw the Repository demonstrated by our team, he commented that identifying clinical scenarios in such a repository could become the basis of certifying a health system's ability to alleviate such problems. Our code-dissemination project using SourceForge is evidence of our continued commitment to sharing and outreach in order to support research elsewhere. Our work on the ICE External Interface is identifying opportunities to connect with other interesting work – the ability to "see, do, share."

These funded activities are examples of significant national healthcare challenges, and the TATRC funding allows us to build the necessary bridges with agencies to address these challenges. We are showing how to operationalize our work, which is not about some distant future but is translational and technology research with near-term applications. We are eager to have more collaborative opportunities.

References

1. Goldman JM, Jackson JL, Whitehead SF, Rausch TL, Weininger S, "The Medical Device 'Plug-and-Play' (MD PnP) Interoperability Program," part of Schrenker RA, "Software Engineering for Future Healthcare and Clinical Systems," *IEEE Computer,* April 2006.
2. Goldman JM, "Medical Device Connectivity for Improving Safety and Efficiency," *American Society of Anesthesiology Newsletter* 70:5, May 2006.
3. Goldman JM, "Patient-Centric Networked Medical Device Interoperability," part of Dagalakis NG, "Report on the Results of the NIST Medical Devices Metrology and Standards Needs Workshop," November 2006. http://www.nist.gov/el/isd/upload/USMS_Med_Dev_Needs.pdf
4. Goldman JM, Whitehead S, Weininger S, "Eliciting Clinical Requirements for the Medical Device Plug-and-Play (MD PnP) Interoperability Program," *Anesthesia & Analgesia* 2006;102:S1-54.
5. Carr S, "Plug and Play for Patient Safety," *Patient Safety & Quality Healthcare*, July-Aug 2007. http://www.psqh.com/julaug07/editor.html
6. Rausch T, Jackson JL, "Using Clinical Workflows to Improve Medical Device/System Development," *Proceedings of the Joint Workshop on High-Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS / MD PnP 2007), Cambridge, MA, June 25-27, 2007*, pp. 133-134. IEEE Computer Society Press, 2008.
7. Schrenker RA, "Ensuring Sufficient Breadth in Use Case Development: How Should Non-Functional Requirements Be Elicited and Represented?" *Proceedings of the Joint*

Workshop on High-Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS / MD PnP 2007), Cambridge, MA, June 25-27, 2007, pp. 135-136. IEEE Computer Society Press, 2008.
8. Cortés P-A, Krishnan SM, Lee I, Goldman JM, "Improving the Safety of Patient-Controlled Analgesia Infusions with Safety Interlocks," *Proceedings of the Joint Workshop on High-Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS / MD PnP 2007)*, *Cambridge, MA, June 25-27, 2007*, pp. 149-150. IEEE Computer Society Press, 2008.
9. Arney D, Goldman JM, Lee I, Llukacej E, Whitehead S, "Use Case Demonstration: X-Ray/Ventilator," *Proceedings of the Joint Workshop on High-Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS / MD PnP 2007), Cambridge, MA, June 25-27, 2007*, p. 160. IEEE Computer Society Press, 2008.
10. Whitehead SF, Goldman JM, "Getting Connected for Patient Safety," *Patient Safety & Quality Healthcare* 5:1, Jan-Feb 2008.
11. Chiao JC, Goldman JM, Heck DA, Kazanzides P, Peine WJ, Stiehl JB, Yen D, Dagalakis NG, "Metrology and Standards Needs for Some Categories of Medical Devices," *J. Res. Natl. Inst. Stand. Technol.* 113, 121-129, 2008.
12. Whitehead SF, Goldman JM, "Hospitals Issue Call for Action on Medical Device Interoperability," *Patient Safety & Quality Healthcare* 6:1, Jan-Feb 2009.
13. Arney D, Goldman JM, Whitehead SF, Lee I, "Synchronizing an X-ray and Anesthesia Machine Ventilator: A Medical Device Interoperability Case Study", *Proceedings of BioDevices 2009*.
14. Whitehead SF, Goldman JM, "Connectivity to Improve Patient Safety: Making Medical Device 'Plug-and-Play' Interoperability a Reality," *Patient Safety & Quality Healthcare* 7:1, 26-30, Jan-Feb 2010.
15. Kowalczyk L, "MGH death spurs review of patient monitors," *The Boston Globe*, February 21 2010.
16. Kowalczyk L, "'Alarm fatigue' linked to patient's death," *The Boston Globe*, April 3 2010.
17. Arney D, Pajic M, Goldman JM, Lee I, Mangharam R, Sokolsky O, "Toward Patient Safety in Closed-Loop Medical Device Systems," In: *Proceedings of International Conference on Cyber Physical Systems*, April 2010.
18. Saver C, "Integrating Devices for Patient Safety," *OR Manager* 26:6, 21-24, June 2010.
19. Goldman JM, Schrenker R, Melendez L, Hampton R, Driscoll W. Implications of the New FDA Medical Device Data System (MDDS) Regulation for Automated Clinical Documentation. *Proceedings of American Society of Anesthesiologists*: Equipment, Monitoring and Technology. October 2011.
20. Arney D, Goldman JM, Bhargav-Spantzel A, Basu A, Taborn M, Pappas G, Robkin M. Simulation of Medical Device Network Performance and Requirements for an Integrated Clinical Environment. Biomed Instrum Technol 2012 Jul-Aug;46(4):308-15. http://mdpnp.org/publications_UN9U.html

Appendices

1. Data Logger Requirements
2. Data Logger Software Architecture: ICE Data Logger Modes
3. Clinical Scenario Repository Slides
4. Code Release Plan
5. HIMSS13 Report

Category	Requirement Number	Requirement	Rationale
Goals			
	G1	The purpose of the data logger is to record low-level device data (e.g. button presses and physiological data values) from individual medical devices, along with location information and data about the status of the medical device network, in an open, standardized, and time-synchronized manner.	It is impossible to trace back to the origins of interactions between devices that can cause serious hazards to patients without a coordinated, time-synchronized log of all of the data sent by all of the devices in the system. This complete data record offers a more complete event picture than the highly filtered and processed data that goes into the EHR.
	G2	As medical device interface capabilities improve, more device data will be available to the data logger.	
	G3	The event recorder will be useful for analyzing adverse events and near misses with patients, as well as debugging interactions between multiple medical devices (such as bedside monitors and remote alarm systems) or between medical devices and other IT systems (e.g. the EHR). We anticipate that this data will also be extremely useful for developing advanced clinical algorithms and analyzing patient outcomes.	
	G4	One function of the playback and analysis software is to assist clinicians in categorizing adverse events. FDA CDRH uses event problem codes and evaluation codes to classify the device problems associated with an adverse event. These codes are harmonized with ISO TS 19218 and there are plans to integrate these codes into SNOMED and to work with IEEE 11073 to incorporate the codes into these two codes sets to create a global vocabulary to report device problems.	
	G5	We anticipate that the log from the recorder will be an important legal record as well as a clinical and engineering tool. This means that the data in the record must be trustworthy, and any tampering with the record must be obvious.	
Types of Logging			
	1	ASTM F2761-09 requires logging of “user interaction with devices” – e.g. button presses – that will help add context to events to facilitate analyses of usability problems.	
	2	The data logger shall support multiple logging modes.	
	3	The data logger shall support a "clinical data logging" mode.	
	4	In "clinical data logging" mode, the data logger must record all "clinical" data communicated through the ICE network controller.	
	5	"Clinical data" must include patient data, commands sent to the devices, button-press reports from devices, and device association data (including ICE app startup and shutdown).	
	6	"Clinical data" may include other data elements.	

Category	Requirement Number	Requirement	Rationale
	7	Optional "clinical data" elements to be recorded may optionally be communicated to the data logger by an ICE application at the time of the application's startup. This is an area of research with additional requirements TBD.	
	8	The data logger shall support a "technical logging" mode.	
	9	In "technical logging" mode, the data logger must record all "technical data" communicated through the ICE network controller.	
	10	"Technical data" must include all clinical data plus each data request, the entire response to requests, and all acknowledgments.	
	11	The data logger must support a "ridiculously detailed" mode.	
	12	In "ridiculously detailed" mode, the data logger must record literally every bit of data communicated through the ICE network controller for a minimum of 30 seconds.	This is for debugging network level problems. The 30 second time period may be adjusted, but is meant to be long enough to capture an event but short enough that the transmitted data can be stored in RAM since it may be coming in too quickly to write it to disk. (Network bandwidth is likely greater than disk I/O bandwidth.)
	13	The data logger may support additional modes.	
	14	Data compression algorithms may be used to reduce the size of the log files, provided that they do not lose data in the process.	This may be controversial - the argument against lossy compression is simply that we cannot anticipate all future uses, and the data once lost can never be recaptured.
	15	The data logger must record raw network traffic even when it cannot interpret the contents.	
	16	Each data transmission from a device must include the unique device identifier (UDI) as specified by the FDA, a logical timestamp as described above, and the data.	
	17	Existing adverse event ontologies should by preference be used, such as the device problem and evaluation codes of the FDA's MDR system.	This isn't really a requirement as written, but will be strengthened as we choose an ontology or combination of ontologies for use.
Timestamps and Logical Clocks			
	18	The network controller must contain a real-time clock set using the network time protocol (NTP).	
	19	A record of synchronization of the network controller clock, and information about the accuracy with which it was set, must be entered in the log as it happens.	
	20	Data from devices on the network must be entered in the log, along with a unique, incremental sequence number.	
	21	Data from devices on the network must be entered in the log, along with a timestamp from the network controller clock.	

Category	Requirement Number	Requirement	Rationale
	22	Each entry in the log must have a cryptographic hash attached.	This facilitates detection of manipulation of log contents.
	23	The data logger must record any timestamps contained in messages being logged as received without modification. It will record them as received and append its own timestamp.	
	24	Logical clocks such as Lamport clocks or a vector clock timestamp may be transmitted by devices. Any such timestamps must be logged with the message.	These timestamps may be useful for debugging some kinds of complicated interactions. They can be used by the playback applications if available.
Security and Trustworthiness of the Log			
	25	Each log entry must include an individual sequence number, a cryptographic signature, and a timestamp recording when the message was received at the network controller.	The sequence number makes it obvious if a record is missing from the sequence, and the signature allows verification that the content of the record has not been changed.
Analysis and Replay of Log Data			
	26	The replay program must be able to open the data log from the event recorder, check it for consistency by examining the signature of each entry, and provide the user with a set of tools for analyzing the data.	
	27	The playback program must support at least two modes. We call the first use "clinical log playback" and the second use "debugging playback".	The log serves two general purposes: it will support analysis of adverse events involving multiple devices and it will allow system developers to view low-level data for debugging their applications. These purposes require different playback tools and techniques.
	28	The playback program must allow analysts to build an interactive composite timeline of logged data and events.	
	29	The playback program must allow the user to link text from clinician interviews in the appropriate places in the composite timeline.	
	30	The playback program must automatically include location information in the timeline when it is available in the record.	
	31	Users must be able to choose between viewing the sequential data stream from a specific individual device and the interleaved sequences from multiple devices.	This should be as "intuitive" and "user-friendly" as possible. Some testing with novice users will be necessary.
	32	In addition to the textual display, the playback program must be able to display a graphical timeline of data values and events from the devices.	
	33	The playback program must allow analysts to display narrative text beside the logged data, tag sections of the entries with times, and mark entries in the graphical timeline.	Clinician narratives are an important part of the adverse event analysis process.
	34	The user must be able to play back the data, either in real-time or at increased or reduced speeds.	

Category	Requirement Number	Requirement	Rationale
	35	The playback program for debugging will allow these users to output data to a file for external analysis. Users must be able to select which data they want and pick an output format.	Debugging playback typically involves too much data to view graphically. If system developers want to see a graphical display, they can use the clinical playback program or graph the data in another application. We expect that system developers will use standard tools like Matlab and protocol analyzer software to examine the files, and we will support this by exporting the data in appropriate formats.
	36	The debugging playback tool must also support down-sampling the data.	This will reduce the size of the files and the strain on the external analysis tools.

ICE Data Logger Modes

History:

Date	Author	Revision number
11/27/2012	Telecon NIST & CIMIT	
12/18/2012	Telecon NIST & CIMIT	
01/25/2013	Telecon NIST & CIMIT	
01/28/2013	NIST (AF, LD)	1.0
01/30/2013	NIST (AF, LD)	1.1
02/07/2013	NIST (KS)	1.2
02/11/2013	NIST (AF)	1.3
02/20/2013	NIST (AF)	1.4
02/25/2013	NIST (AF)	1.5
03/05/2013	NIST (AF)	1.6
03/10/2013	NIST (KS)	1.7
03/12/2013	Telecon NIST & CIMIT	
03/12/2013	NIST (AF)	1.8

Appendix 2 – Preliminary Software Architecture Document

1 Goal

During the process of identifying the requirements for the ICE [2] data logger, it became necessary to further investigate the logging modes definition and implementation.

The goal of this document is to

- Define the different modes that the ICE data logger may choose to operate under
- Clarify the association of data with logging modes.
- Explore approaches and implementation strategies that can support logging modes in the ICE environment.
- Determine the impact on the data logger requirements.

2 Introduction

In order to allow users, including clinicians, developers, or system engineers, to select the level of details in the logged data, a logging system may include various options, hereafter referred to as “modes”. Each mode will enable different use case scenarios as will be discussed later in this document.

Many data types can also be identified. These include ‘high-level ICE messages’ such as the one used to send a command to a medical device from an ICE app to ‘low-level messages’ such

as the packets used by a transport layer to carry the data between a MD device and the ICE Network controller.

A goal of this document is to discuss different modes that the ICE data logger may support and their intended use. Use cases supporting the modes will be introduced. Based on the identified modes and their intended use, it should then be possible to classify the data types and their association to various modes.

This effort will hopefully clear the path to develop a technical approach that will support various modes of the ICE data logger.

3 Data Logging Modes Definition

The following two hierarchical data logging modes have been identified in [1].

- Clinical
- Technical/troubleshooting

In order to determine what data types are associated with each mode, it is necessary to understand the purpose of each logging mode, or more precisely the use-cases that are enabled by each mode.

3.1 Mode: Clinical

Clinical mode: the minimum amounts of data that can be recorded to enable the following use-cases constitute the Clinical logging mode.

> **Medical Use-case:** Analyzing adverse events and near misses during a medical procedure. Document to enable reporting of devices in use, time of use, duration of use, and error codes – even if they did not result in device malfunction.
> **User:** clinical staff, clinical engineers, other non-clinical staff in clinical environments
>
> **Research & Educational Use-case:** developing advanced clinical algorithms and analyzing patient outcomes, identifying best practices or common mistakes.
> **User:** clinical staff, clinical researcher, academics, students
>
> **Quality, Safety, and Legal Use-case:** Used by hospital safety and quality improvement office, insurance carriers, or in a court of law to identify a failure by medical personnel during a medical procedure or identify device/app and their corresponding hardware/software failure or malfunction that led to an adverse effect during a medical procedure.
> **User:** clinical and legal experts, IT-experts, biomed experts

3.2 Mode: Technical

Technical mode: the minimum amounts of data that can be recorded to enable the following use-cases constitute the Technical logging mode.

> **System Integration and Medical device interaction**: Debugging malfunctions resulting from the interaction between multiple medical devices or a medical device and other IT systems (during deployment, setup, or system integration)
> **User**: System Engineers, Device manufacturers
>
> **ICE App development & Debug Use-case:** Used to debug ICE apps during and possibly after development phase.
> **User:** Medical app developers

More use cases may be identified and added to the above lists.

Each mode is a subset or superset of another mode e.g. data recorded in the Clinical mode is a subset of the Technical mode logged data.

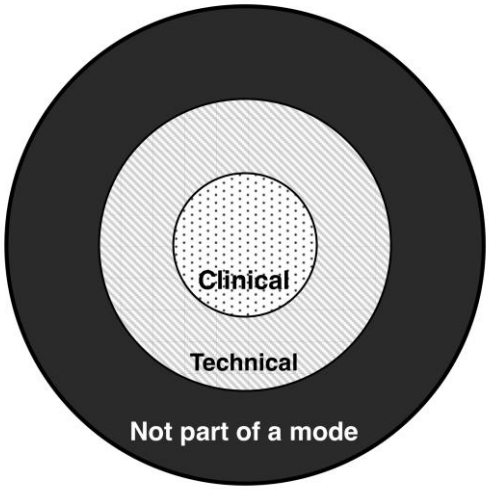


Figure 1: Clinical mode is a subset of the technical mode, which is a subset of all the available data.

3.3 Customizing modes

The notion of minimum amount of data can be seen as the baseline of a logging mode. A logging mode may actually contain more data than required by the baseline but cannot have less. A clinician may wish to have more data logged compared to the default amount of data that each mode provides. Assuming that the system can handle the load introduced by logging the extra data. Both logging modes and customization can be used together as long as the customization doesn't break the baselines of the modes.

4 Minimum functionalities required by the data logger

Having several logging modes introduces more complexity: a single mode means there is no need to categorize messages. The ICE data logger would need to filter the data because logging everything may neither be feasible nor useful.

The concept of logging modes implies that the system must have mechanisms to categorize and filter the messages:

- **Categorization** consists in assigning a type (or label) to a single message or a single piece of information.
- **Filtering** consists in creating multiple sets from a single set based on some criteria.

These 2 mechanisms (i.e. categorization and filtering) might be merged in a single component/module.

It should be noted that categorization and filtering would be required in order to implement the data logger whether it uses logging modes, logging customization or both.

The remainder of this section describes categorization and filtering; and explores the implications of having them merged versus separated.

4.1 Categorization and filtering merged

When the two functionalities are merged, there might be no need to tag the message because the filtering is done within the same component. (example?) As soon as the category of the message has been established, it can be either recorded or discarded.

Hub components, that receives all data, seem to be the most appropriate components to perform the filtering. In the ICE, the network controller is a very good candidate. The ICE data logger could also assume this functionally as long as the network controller forwards all messages transported on the ICE to the data logger.
If the filtering is merged with the data categorization, then this means that the data logger or network controller needs to be able to determine the message mode. In other words, it needs to identify every data type used within the ICE.

Detailed knowledge of all ICE applications may not be available to the ICE data logger or network controller; however, it could be required to provide a complete categorization of all data. So these components don’t seem to be the most appropriate modules to perform the data categorization.

4.2 Categorization and filtering separated

It has already been mentioned that the filtering function can only be performed by a hub component since it has access to all messages passing through the system.

The best way to guarantee that all data will be properly categorized is to have it done by components that are able to interpret or understand the messages and their content. These components are the ones that produce the messages (e.g. medical device). Some intermediary components may be able to perform interpretation but due to the layered architecture of the system and communication stack or the use of encryption or compression algorithms, it might not be feasible for such intermediary components to perform this functionality.

5 Possible approaches to categorize data

This section explores different approaches that could be used to associate ICE data with a logging mode, in other words, perform the data categorization.

Each approach uses different characteristic of the data to categorize them. The characteristics used are the protocols supporting the data, the OSI layers of these protocols, the data types or the data type instances. We also assess the benefits and drawbacks introduced by each approach.

The following diagram (Figure 2) is an example highlighting potential protocols that could be used in the ICE. The diagram shows protocols based on their position in the OSI model and doesn't represent potential protocol encapsulations.

Appendix 2 – Preliminary Software Architecture Document

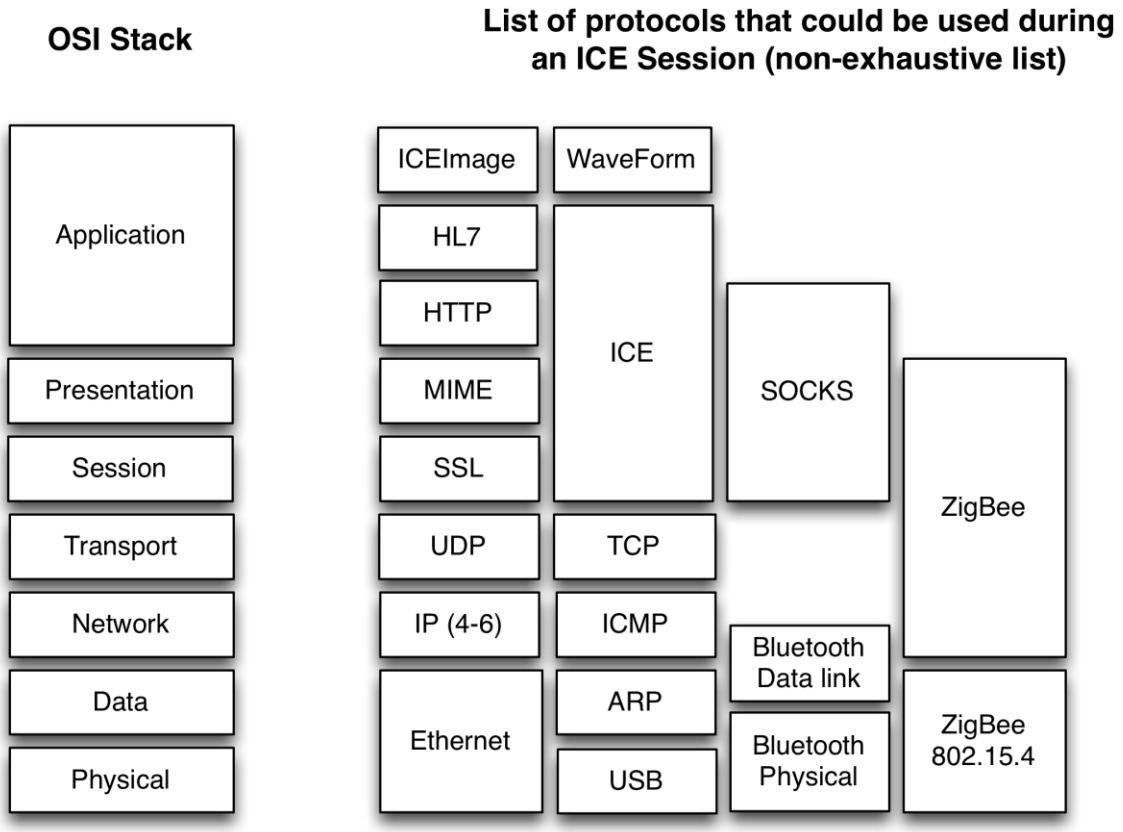


Figure 2: Potential protocols that could be used in an ICE.

5.1 Approach 1: Using the OSI model

An abstraction model such as the OSI model could be used to define logging modes. The following diagram (Figure 3) highlights an example of partitions.

Appendix 2 – Preliminary Software Architecture Document

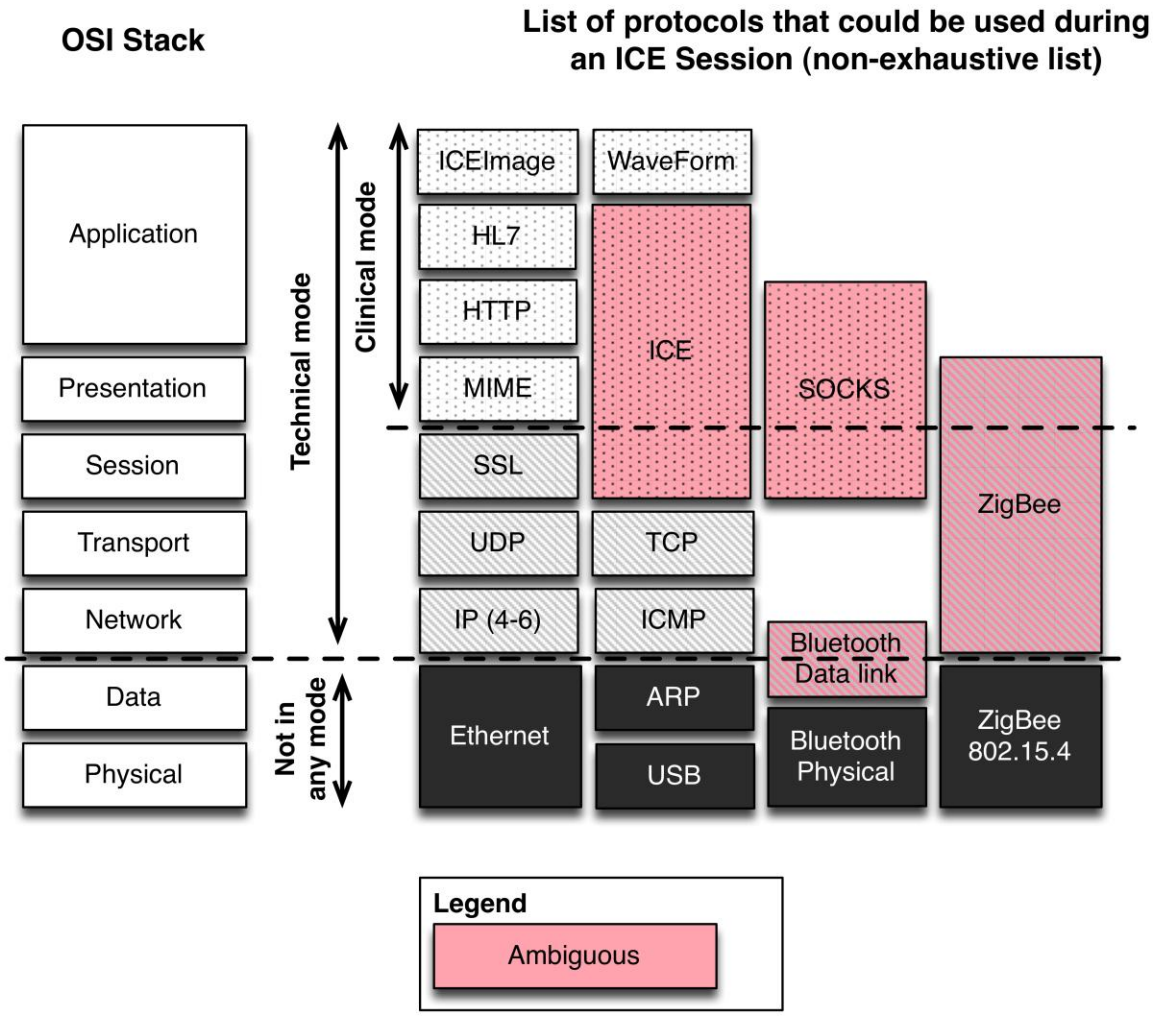


Figure 3: ICE logging modes are defined using the OSI layers.

In this example, any message or packet that belongs to a protocol sitting at level 5 (session) of the OSI stack or below would be considered technical data and thus belong to the technical mode. Data from protocols sitting on the level 6 or above of the OSI model would fall under the clinical mode.

This partitioning may not be precise enough: some protocols or drivers (such as the Zigbee specifications, SOCKS or the ICE communication protocol) belong to several layers and therefore it is not clear if their messages should be categorized as technical or clinical.

If such an approach were used for message categorization, a protocol detector would be required to detect the message's protocol in order to identify to which layer they belong. It might be difficult for the data logger to determine which layer a message belongs if the protocol is unknown to the data logger.

A partitioning based on the layers of the OSI model could be used for categorization. However, the categories defined using the OSI layers may lack the precision to support the logging mode of the ICE.

5.2 Approach 2: Using protocols.

Another approach would categorize messages based on the protocols used.

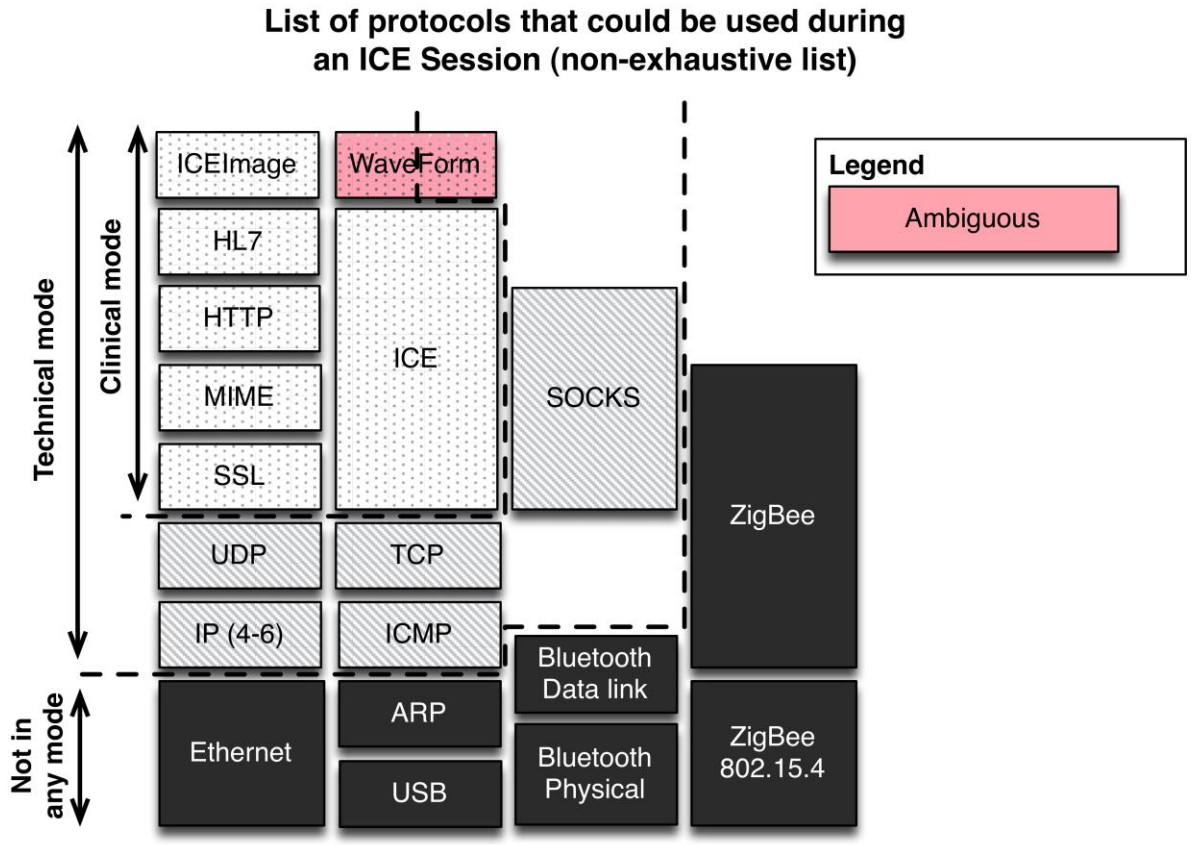


Figure 4: ICE logging modes are defined using protocols.

In the previous diagram, logging modes are defined using protocols rather than the layers of the OSI model. In the diagram, we consider that the ICEImage, WaveForm, ICE, HL7, HTTP, MIME and SSL protocols belong to the clinical logging mode, therefore their messages will be considered as clinical data by the IDL.

This solution offers more granularity for categorization than the approach based on the OSI layers but this may still not be granular enough to define the ICE logging modes.

In our example, we assume the WaveForm has been defined and standardized as a protocol running on top of the ICE communication protocol to transport various information (waveforms, R-R, etc) generated by an ECG and that it is possible to detect messages from this protocol. We also assume we have a scenario that requires logging the R-R but not the waveform.

In that case, the waveform data of the first ECG would be considered technical while the R-R data would be clinical. This approach, which defines modes using protocols, wouldn't be able to categorize a message from the WaveForm protocol that contains both the ECG waveform and the R-R.

Like the first partitioning approach based on the OSI layers, this is still a rough method to categorize messages and data. Although this second approach allows a more detailed categorization of the data, an ICE may require more details to support the scenario mentioned in this section.

An even more detailed partitioning approach may therefore be required.

5.3 Approach 3: Using data types

This approach would not rely on information associated with the data types such as the protocols, or their respective OSI layers to do the categorization. Instead, the data contained in the messages would be used. In order to support this approach, an understanding of all data (including the various ICE device models in use) in the ICE environment would be required to associate each data type to a logging mode. Therefore this approach would support the scenario described in the previous section i.e. the clinician decides to log R-R data but not the waveforms of an ECG device.

However a clinician may need even more granularities for its logging preferences and request to log the waveforms from one ECG device but not the ones from a second device. In that case, the categorization based on the data types could not support the clinician request.

5.4 Approach 4: Using data type instances

In order to have the capability to log the waveforms from one ECG device but not the ones from a second device, the system must be able to differentiate data based not only on their types but also on the data type's instances (when the ICE is in operation).

Therefore knowledge of the various ICE components must be available in order to perform the categorization required to log the data desired by the clinician. The complete knowledge required may only be found within the ICE supervisor. This approach is the most detailed presented so far. An even more detailed approach would be able to pick data to log within a waveform stream for example.

So far, various approaches to perform categorization have been considered. The following section introduces various technical solutions that could associate data with logging modes.

6 Potential techniques to associate data with logging modes

This section explores potential techniques that could be used to categorize data, i.e. associate data with logging modes. The first one uses a detection process to identify the data while the second method uses prior knowledge to categorize the data.

6.1 Approach 1: Detection

Detection could be used to perform the data categorization. The detection engine wouldn't require prior knowledge of the running ICE application but would instead use an engine to identify data and associate them to a logging mode.

An inference engine or a complex processing engine could be used to perform the detection required to categorized data. There is however a risk: no guarantee can be provided that these engines would be able to perform categorization of all data. A new protocol or data type may be introduced in the ICE. If the inference engine has no knowledge about this new protocol, it will fail to identify and categorize it.

The detection approach may introduce another issue. The performance of this process is not deterministic. In other words, it may not be possible to assess how long it will take to identify data because detecting messages from one protocol may be more time consuming than detecting messages from another protocol. If a-priori the proportion of messages between protocols or types is not known, it might be impossible to know how much time will be required to process messages that are sent to the logger.

This may impact a potential functional requirement of the data logger. The requirement states that the data logger must announce its performance capabilities to avoid being overloaded by the system.

It should be noted that if messages are encrypted, the detection process might fail completely.

Using a detection engine could be a solution to perform data categorization. It may not detect every desired data type; therefore, updates of the engine would be required to keep the detection rate high.

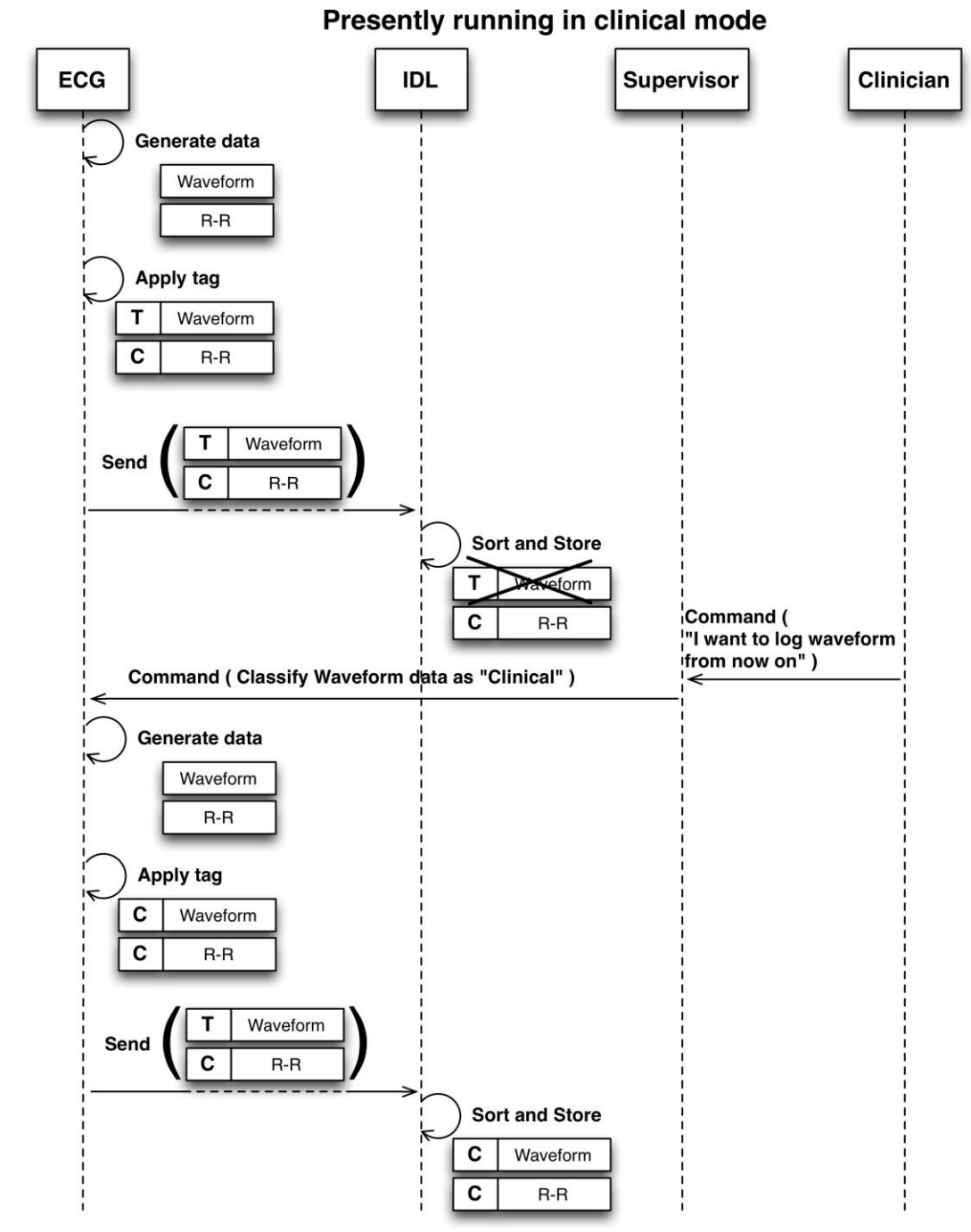
6.2 Approach 2: Categorization using prior knowledge

The previous section explored technique(s) to associate data types to logging modes by using a detection process. Here we look at techniques for categorization that use the knowledge of the ICE application already available from the ICE system.

The following sections explore approaches for categorization performed by devices or by another ICE component.

6.2.1 Categorization performed by the device

The categorization could be done by medical devices as shown in the following sequence diagram.



In this sequence diagram, the device itself associates a logging tag to each data instance. The data logger then determines if the data instance should be logged using this tag.

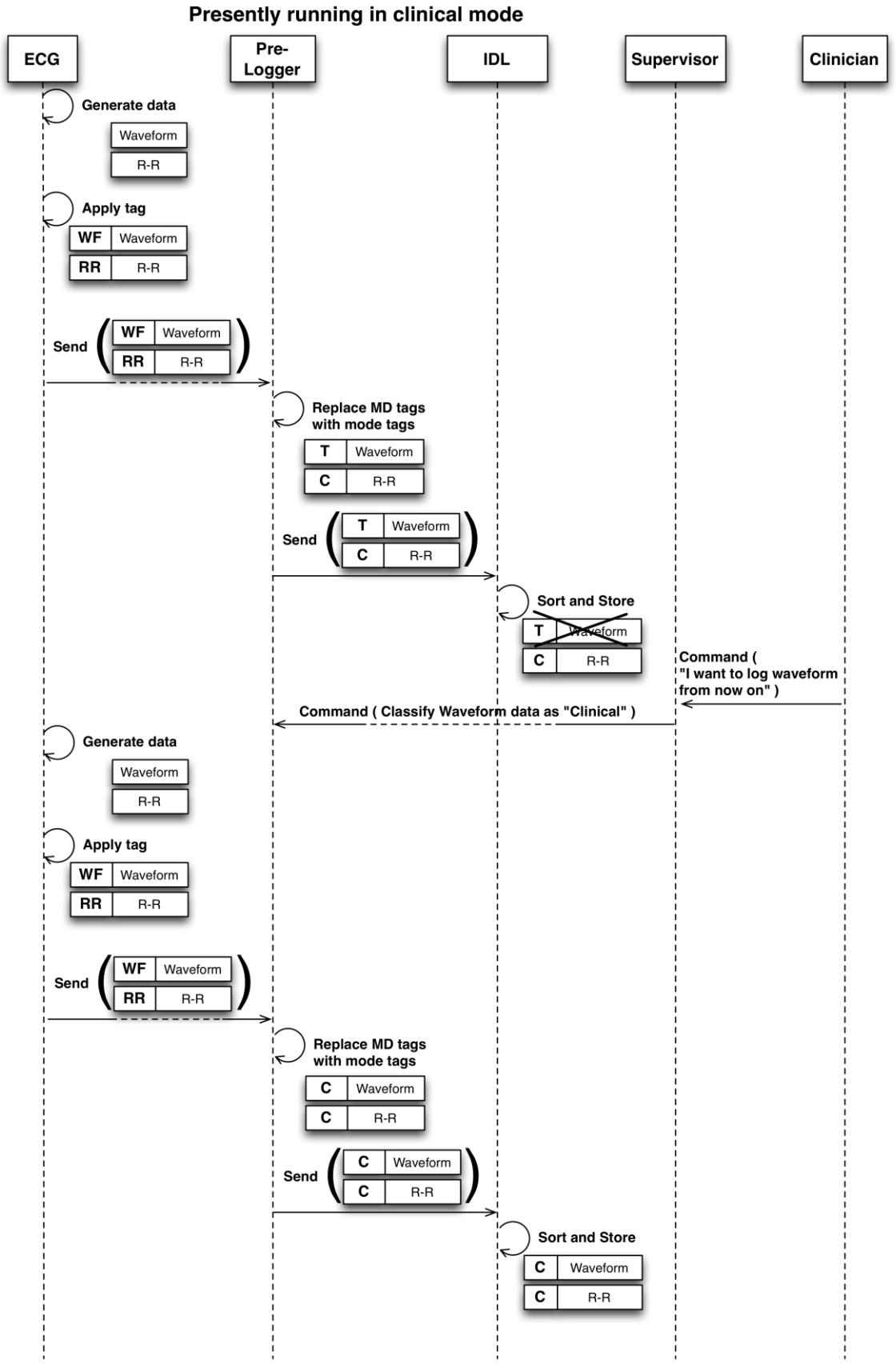
Having such a mechanism will introduce several requirements on some components of the ICE:

- The device needs to be aware of the various existing logging modes, because it is responsible to associate a logging tag to the data instance.
- The device needs to be able to receive and interpret commands to change the logging mode associated to a specific data instance.
- The device needs to maintain a table that stores the logging mode to data instance associations

6.2.2 Categorization not performed by the device

The categorization could be done by another component. In the following diagram, the categorization is performed by a standalone component. This component could be integrated with other existing ICE components. We chose to separate it to clarify its functionality.

Appendix 2 – Preliminary Software Architecture Document



In the above diagram, the association of data instances to logging modes is not performed by the device instead by a component hereafter referred to as “Pre-Logger”. The data logger then determines if the data instance should be logged or discarded using tags.

This mechanism introduces its own set of requirements, including:

- The Pre-Logger needs to be aware of the various existing logging modes, because it is responsible to associate the logging tag to the data instance.
- The Pre-Logger needs to be able to receive and interpret commands to change the logging mode associated to a specific data instance.
- The Pre-Logger needs to maintain a table that store the logging mode to data instance association

A main advantage of this approach is that it shifts the categorization process from the device to the Pre-Logger.

7 Conclusions

- TBA

8 References

[1] David Arney, Sandy Weininger, Susan F. Whitehead, and Julian M. Goldman, “**Supporting Medical Device Adverse Event Analysis in an Interoperable Clinical Environment: Design of a Data Logging and Playback System”**, Proceedings of the *International Conference on Biomedical Ontology (ICBO 2011)*, page 335-339, 2011.

[2] ASTM F2761-09. ASTM F2761-09, New Specification for Equipment in the Integrated Clinical Environment - Part I: General Requirements for Integration.

Clinical Scenarios Repository

MD PnP Program

MD PnP Program

Supported in part by:

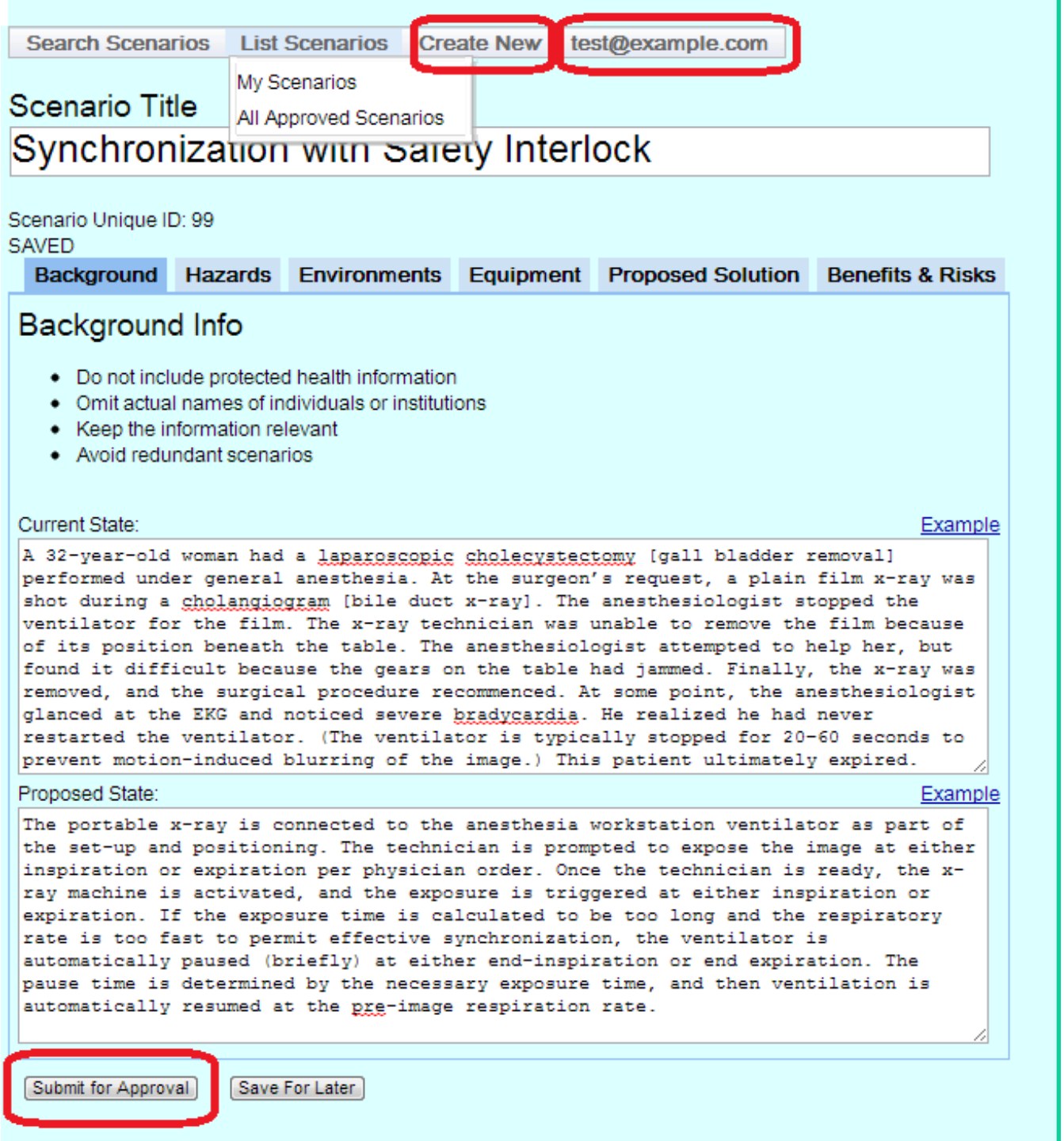
- Department of Defense US Army Medical Research and Materiel Command, award numbers W81XWH-12-C-0154 and W81XWH-09-1-0705
- NIH/NIBIB, grant number 1U01EB012470
- Department of Defense US Army Medical Research Acquisition Activity, award number W81XWH-09-2-0001
- National Science Foundation, grant numbers CNS-10-35715, IIS-1239242, CNS-08-34524, and CNS-08-34709
- National Institute of Standards and Technology, grant number 70NANB10H258

The contents of this presentation are solely the responsibility of the MD PnP Program and do not necessarily represent the official views, opinions, or endorsements of the granting agencies.

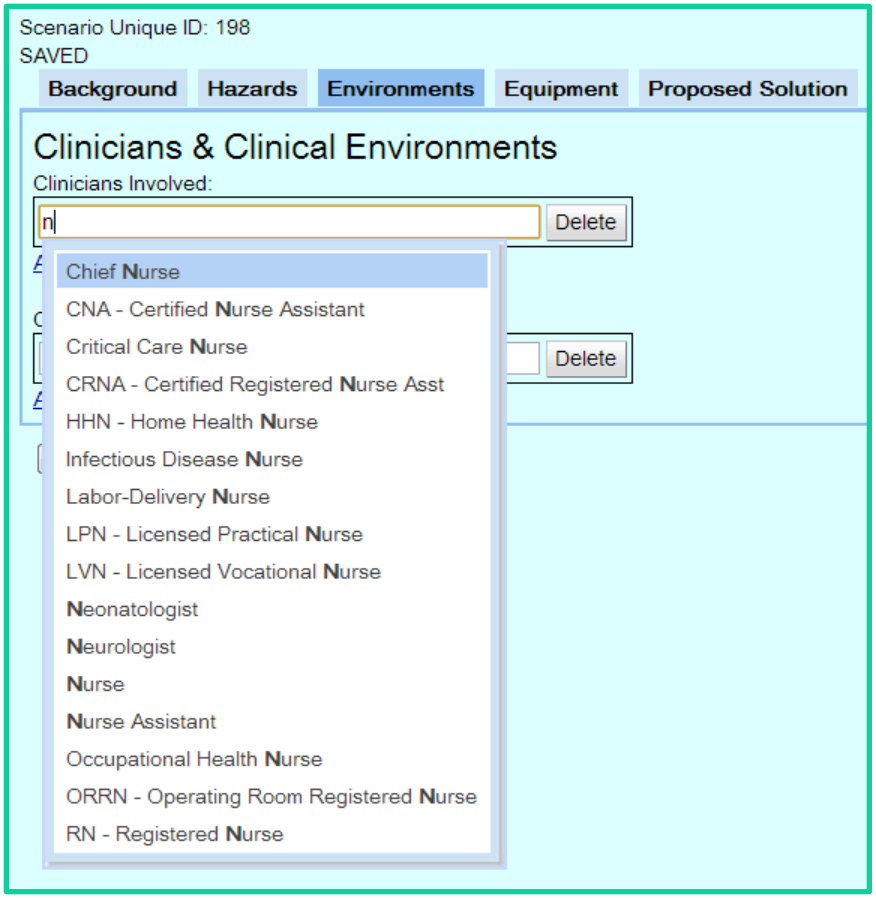
Need for a Clinical Scenario Repository

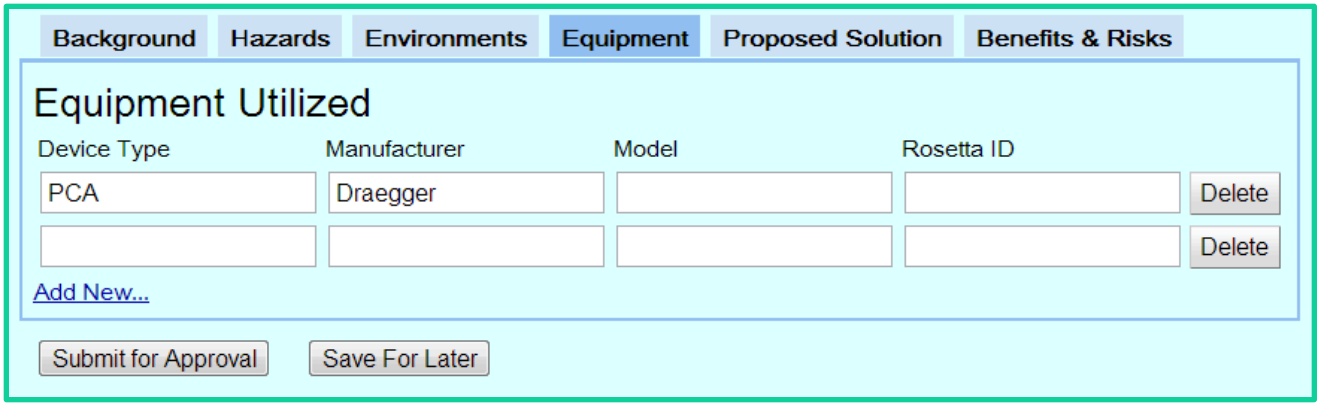
- **Clinical Scenario**: A brief description of a clinical situation or event. The purpose is to inform of the need for development of technical solutions.
- **Clinical Scenario Repository**: A web portal to allow clinicians, clinical engineers and other users to enter, revise and annotate clinical scenarios.

A place to document and share these scenarios will help to identify clinical and technical challenges, address healthcare needs and guide improvements in patient safety and quality of healthcare delivery.

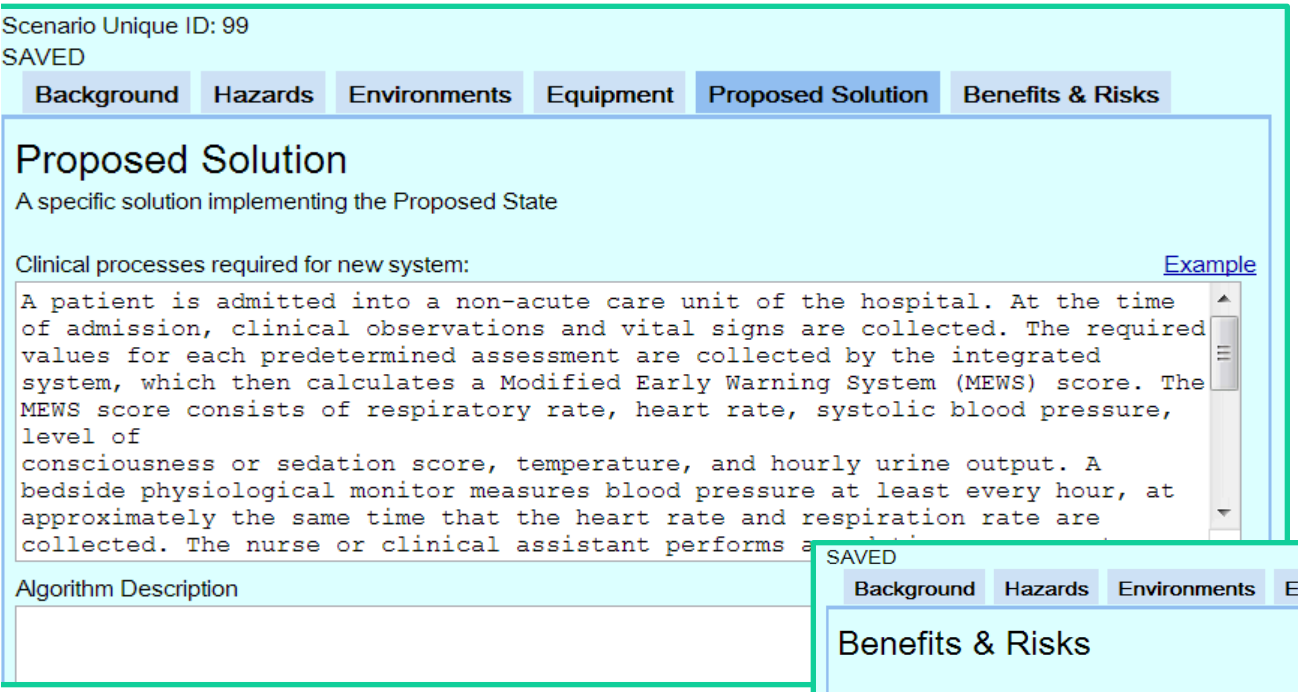


- Instant login / registration process
- Easy way to create a new contribution
- Breadcrumbs approach: users can go back to the tab they missed and not lose any data.
- Save for later and manage your contributions



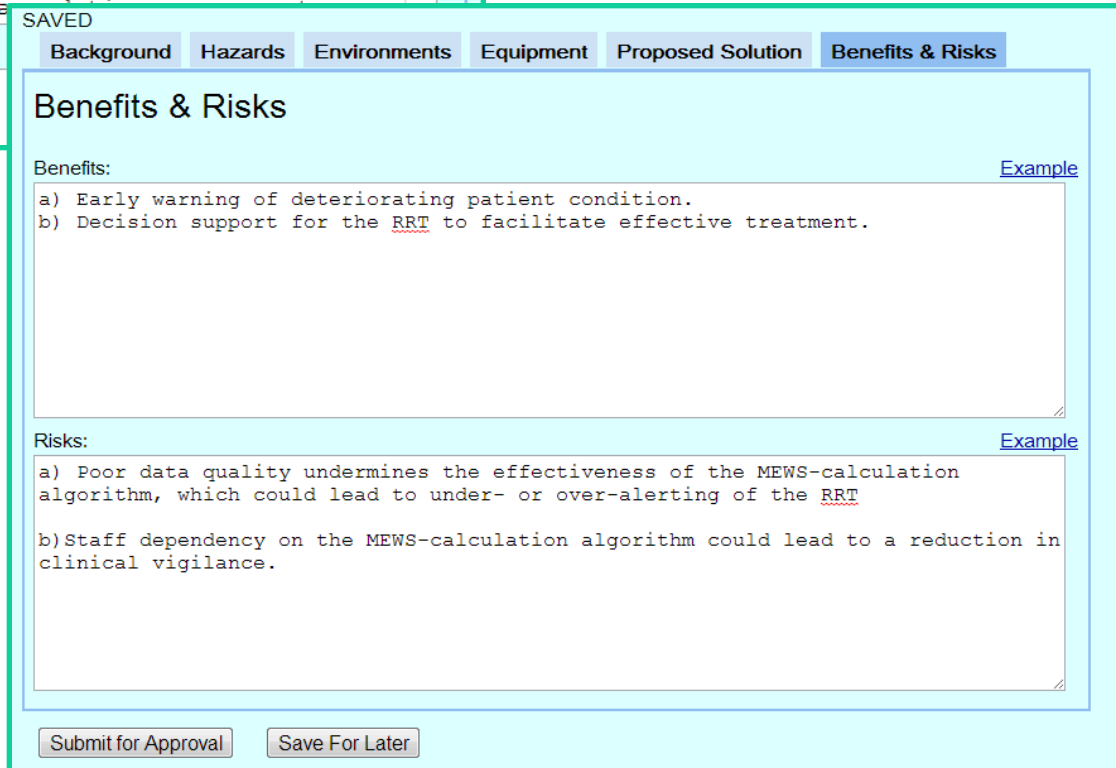


- Enter clinicians, clinical environments and equipment.
- Choose from a preselected array of options, or input your own.

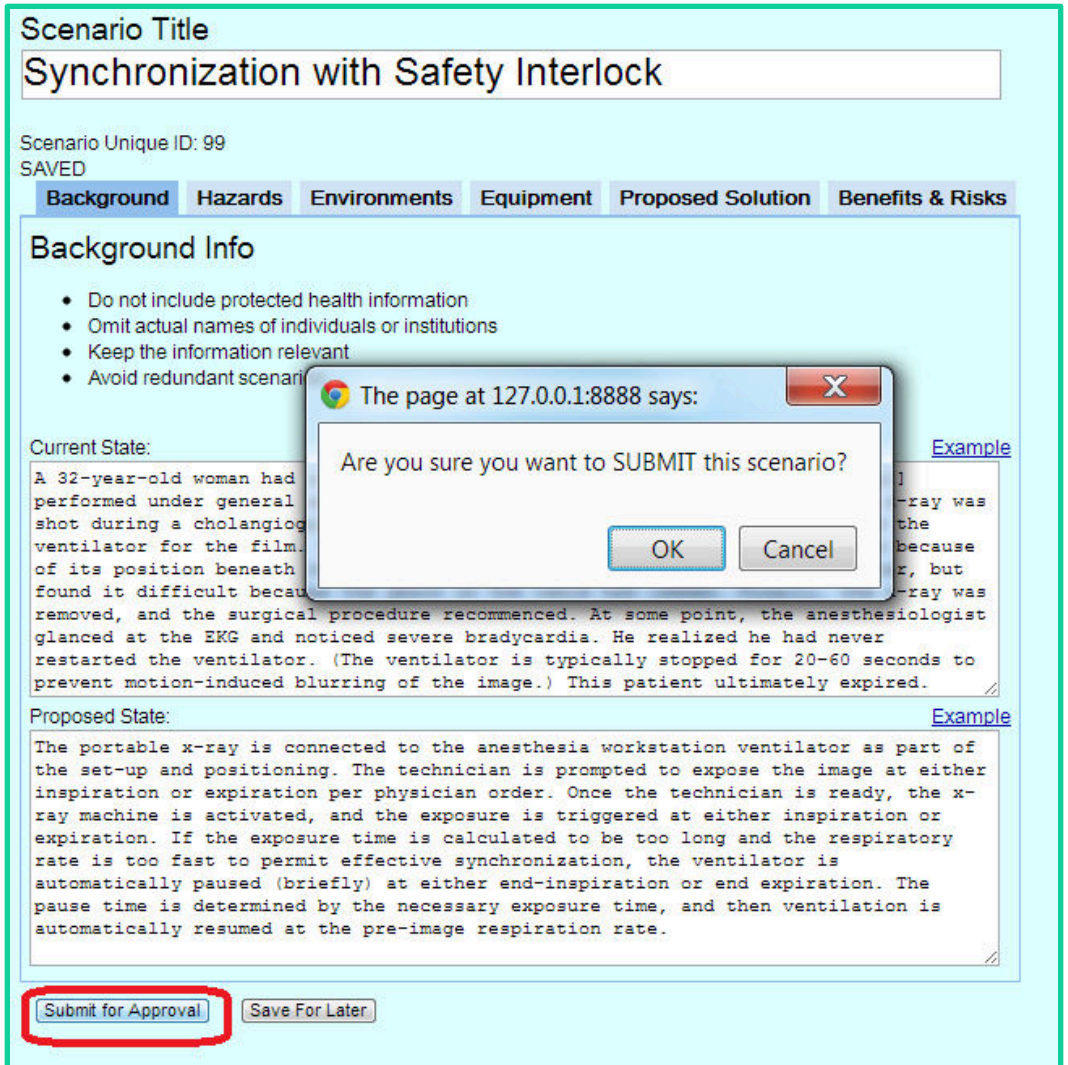


Add a **clinical concept of operations** to show the improvement in safety and effectiveness via a specific solution implementing the proposed state.

Describe the benefits of the proposed process and analyze its potential risks.

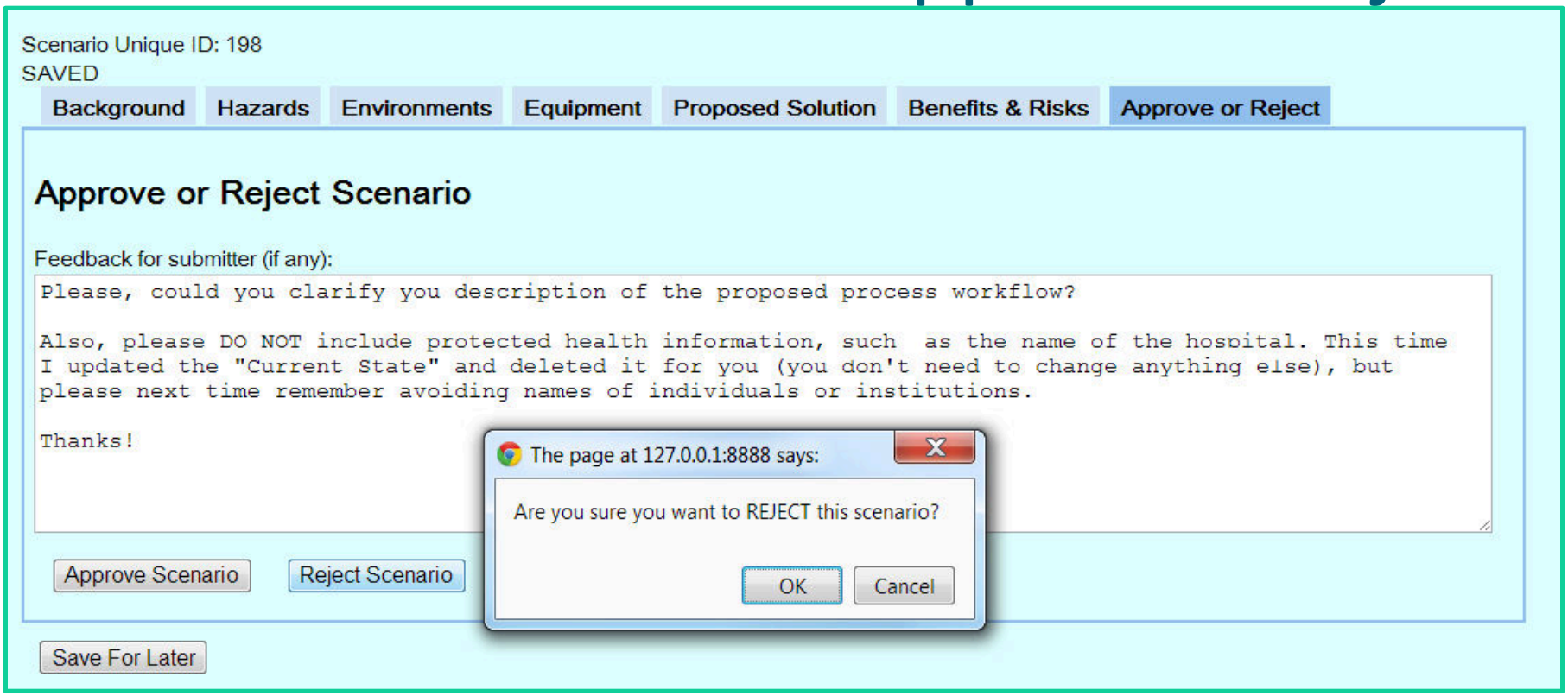


Submit scenarios for revision and approval



- Users can submit a contribution for approval.
- Administrators / reviewers will make sure the scenario meets certain conditions (does not include protected healthcare information), and is relevant for study purpose.

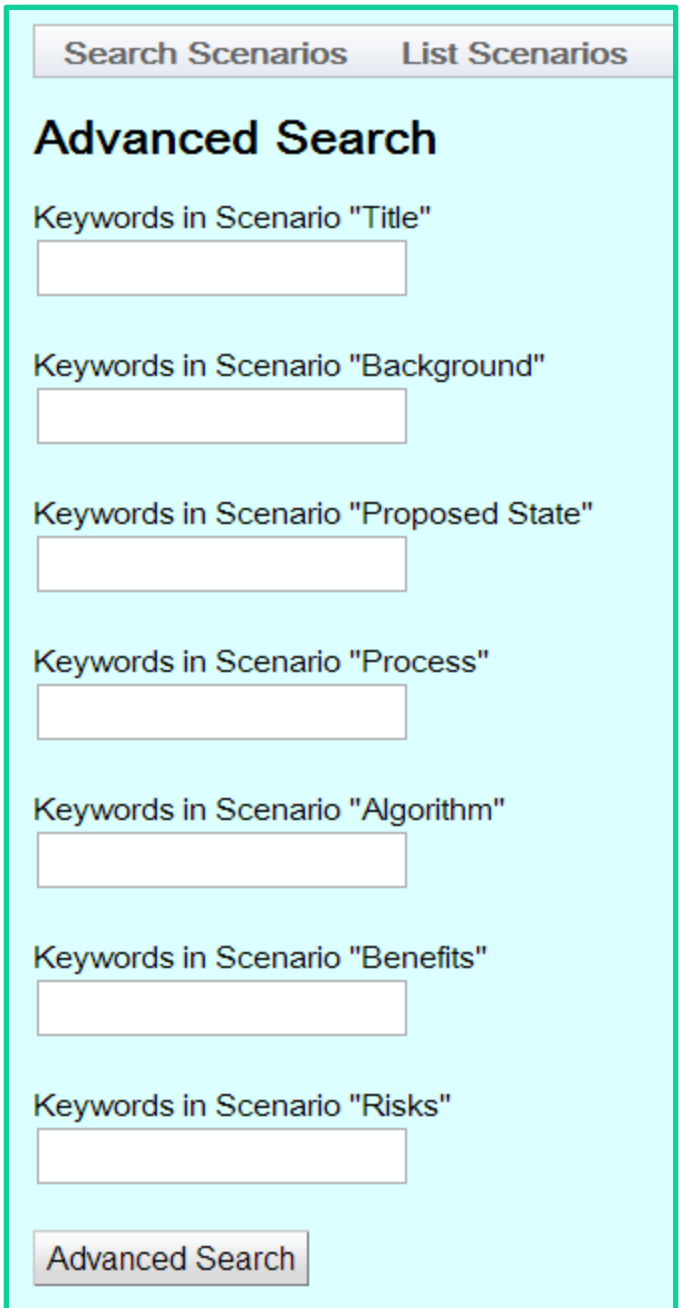
Review the scenarios: approve or reject

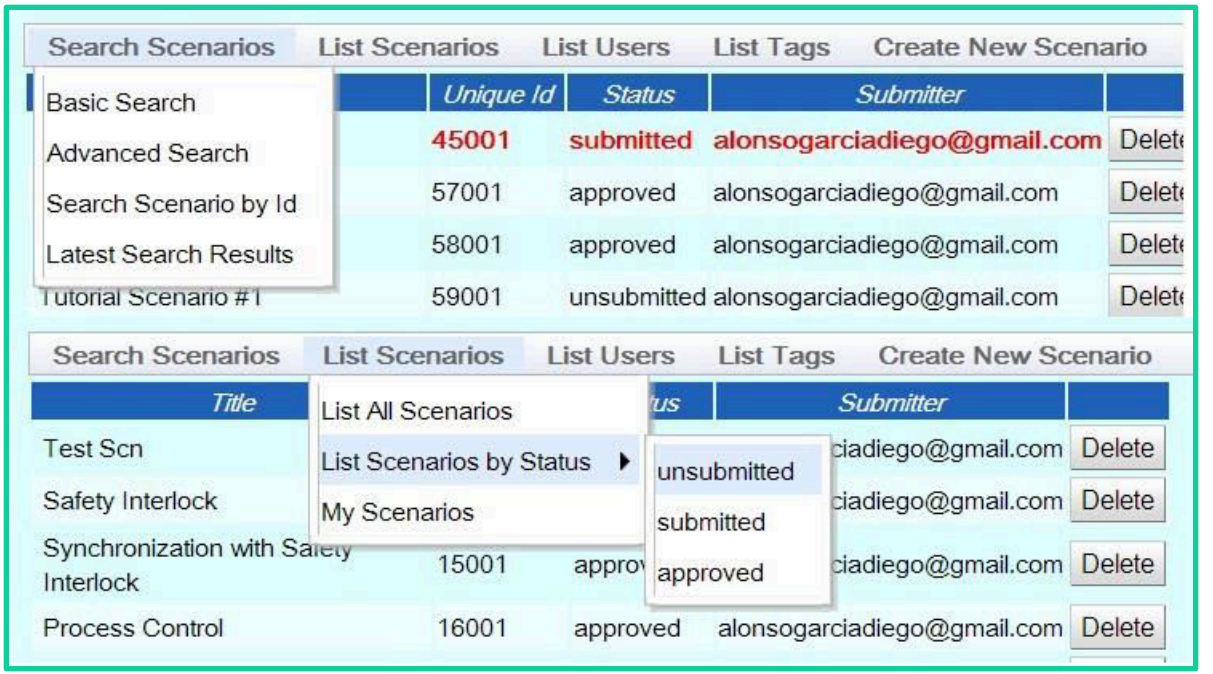


- Approving an scenario makes it available to all users and visitors.
- Users can review, complete and correct their contributions when further clarification has been requested, and submit them again.

Search & List Scenarios

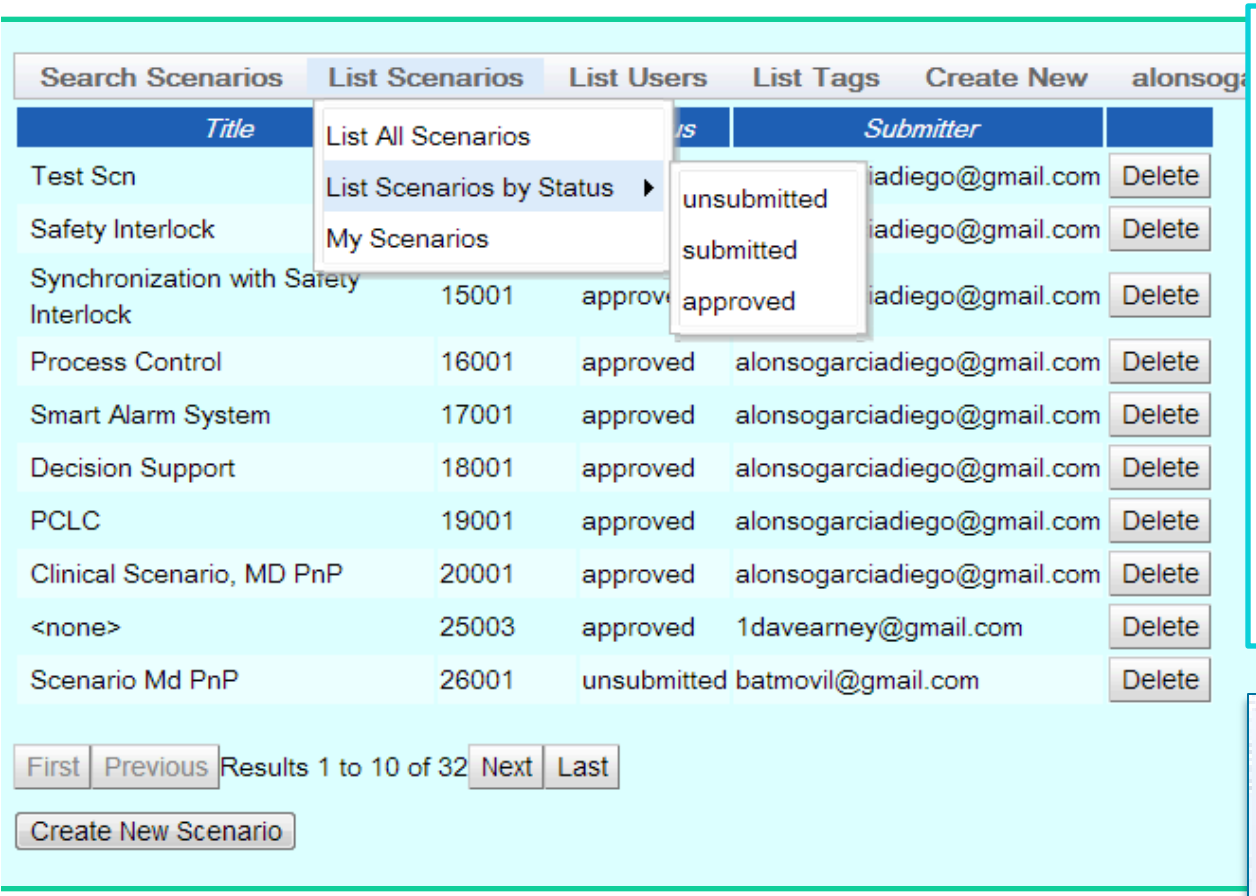
- Basic search & Advanced search: look for scenarios using a specific keyword or with a certain property.
- List scenarios.





Next step: improved advanced search (Date range, submitter, hazards severity, equipment type/manufacturer, clinicians, environment)

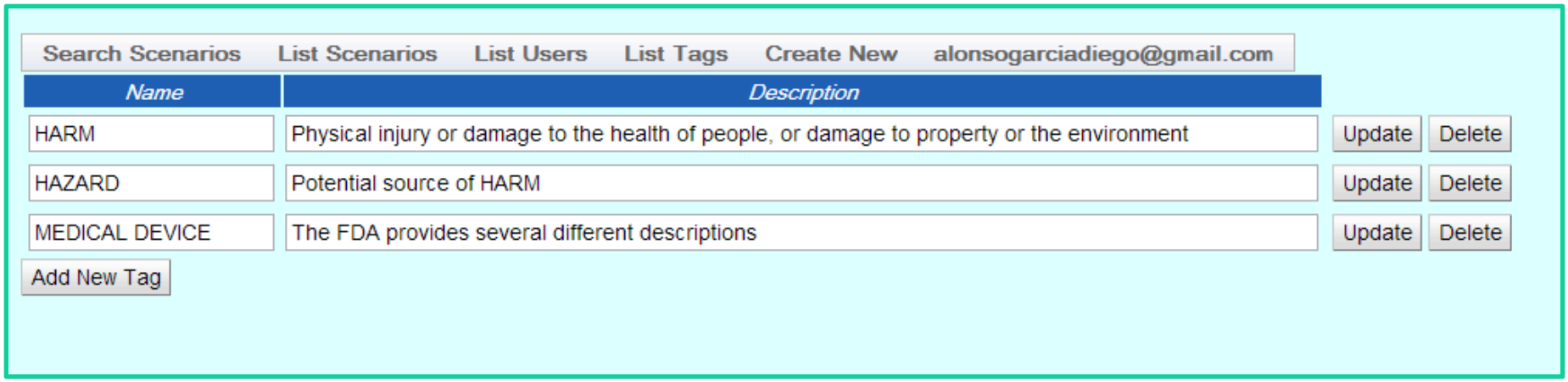
Management of Scenarios



- List those scenarios meeting certain conditions (role–specific features to manage contributions).
- Review *My Scenarios*.

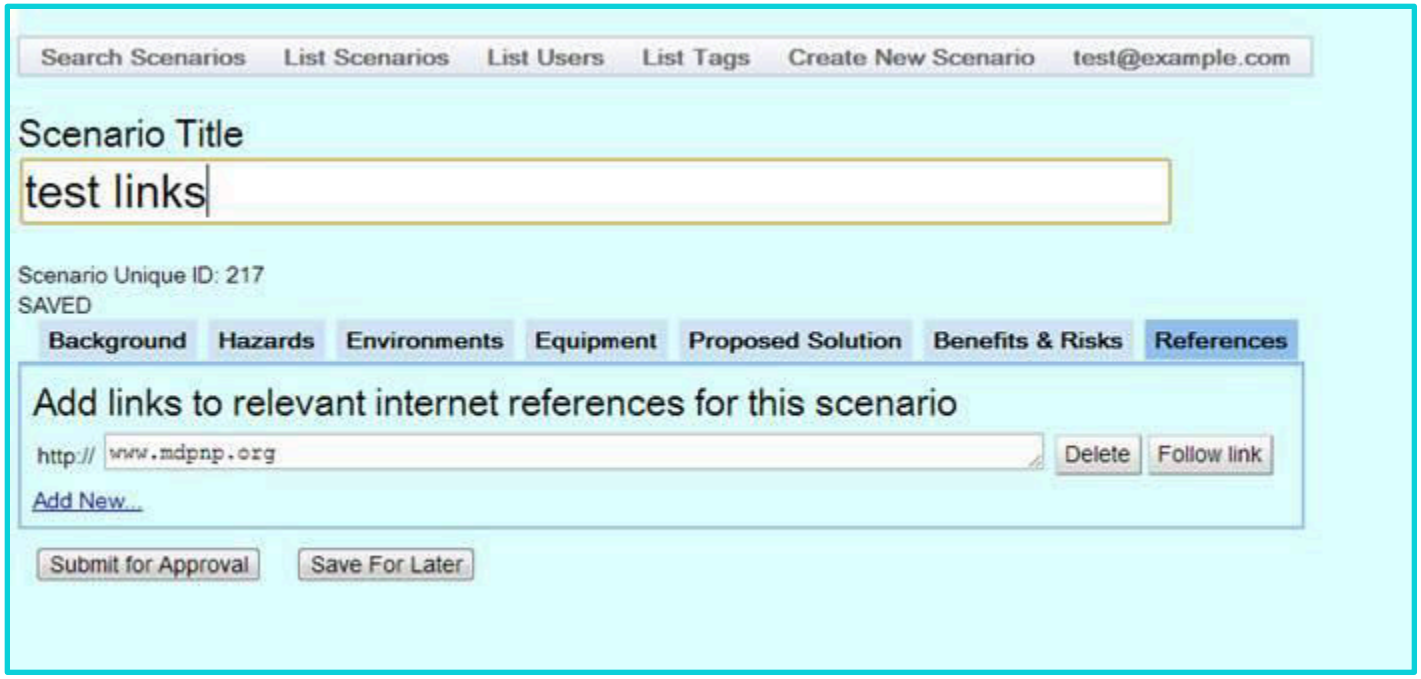
Next: Add descriptive information to track status changes (most recent action taken).

Tag a Scenario



- Administrators will be able to create keywords to tag the scenarios, improving search process and keeping scenario information more meaningful.

Next features



- Add relevant references for the scenario (Considering: text, web references, images, documents…).
- Other ideas: Print/Export a scenario, search by “tag”,…

Appendix 4

Code Release Plan

Currently Released on SourceForge
Latest: http://sourceforge.net/p/mdpnp/wiki/Available/

- **Clinical Scenario Repository**
  - Current "alpha" version in-progress
- **data-types/**
  - The Device Model Working Group is at work in this area.
  - Work (in progress) on data type definitions
    - `x73-idl` Prototype Interface Description (Definition) Language types
    - `x73-idl-rti-dds` Code generation from IDL with rtiddsgen.
- **devices/**
  - Implementation of device protocols for lab devices
  - To be used in conjunction with other components that provide nomenclature and information model translation
  - Current Devices
    - CardioPulmonary Corp Bernoulli
    - Draeger Medibus
    - Masimo Radical-7
    - Nellcor N-595
    - Nonin 9560 (OnyxII w/Bluetooth) / Nonin 3150 (WristOx$_2$)
    - Oridion Capnostream20
    - Philips MP70
- **himss-2013/**
  - Example of submitting a document to a CONNECT 4.0 server
  - Example of creating a custom CONNECT 4.0 adapter to receive such documents
- **interop-lab/**
  - Software demonstrations shared with visitors to the MD PnP Interoperability Lab.
    - *android-apps/*
      - Simple android demos ... currently connects to Nonin bluetooth pulse oximeters
    - *demo-apps/*
      - Demo Applications (currently demonstrates an ICE Supervisor and many ICE Device Interfaces)
    - *demo-devices/*
      - Binding of device protocol implementations (see above) to RTI DDS with IDL defined in data-types/x73-idl-rti-dds
    - *demo-guis/*

Appendix 4

        - GUI components (mostly waveform related) that can be used with any framework (swing, jogl, android, etc)
    - *demo-guis-jogl/*
        - Components to bind demo-guis to Java OpenGL (JOGL)
    - *demo-guis-swing/*
        - Components to bind demo-guis to Swing
    - *demo-purejavacomm/*
        - An implementation of SerialProvider that uses Purejavacomm for serial port access
- **files**
    - Bundled ICE System Distribution
        - A compiled version of the code that's easy to download and run
        - Our PI, Dr. Goldman, was able to download, install, and run in less than 8 minutes.
    - Collected data from Bedmaster and CPC systems illustrating PCA scenario

APPENDIX 5: Report on HIMSS 2013 Demo

The MD PnP / QMDI Team presented a demonstration of a prototype interoperable system at the HIMSS13 conference in March 2013 – the annual meeting of the Healthcare Information and Management Systems Society. This was a great opportunity to show a demo of our QMDI work and get feedback from a diverse audience of engineering, industry, healthcare delivery organizations, military health, and federal agencies. This report summarizes the work presented and lessons learned.

Our HIMSS13 demo venue was a kiosk in an area of the Interoperability Showcase sponsored by the ONC (Office of the National Coordinator for Health IT), which featured numerous projects selected by the ONC as representative of the developing Federal Health Architecture (see Figure 1). Our MD PnP Team builds and publically presents these kinds of demos regularly at HIMSS and other venues, and we find that there are three important aspects that contribute to MD PnP program progress: (1) building the demo system requires each member of the team to contribute their latest developments, so these demos serve as checkpoints for coordinating the work of the program; (2) presenting the demo

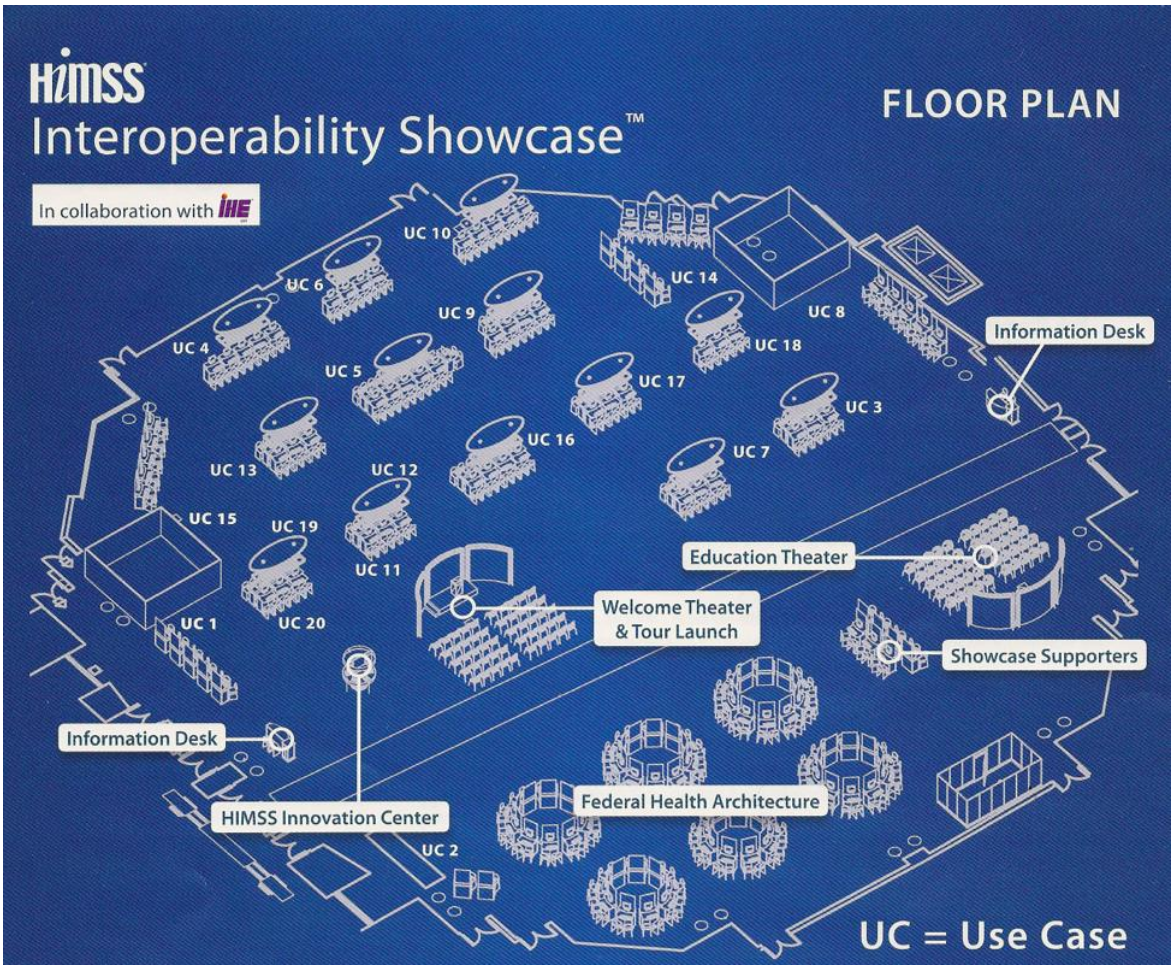


Figure 1: Floor Plan of the HIMSS13 Interoperability Showcase

publically provides an opportunity to get feedback from other experts in the field and from people working in other areas of health IT (HIMSS is an excellent venue to see other work in the broader space

of interoperability and to connect with people who may have heard of our program only indirectly); (3) participating in a HIMSS demo gives us a platform to convey our vision of interoperability, educate the community, and take a leadership role in defining challenges and opportunities for interoperability to make a difference in patient care. This last point was enhanced this year by being invited by ONC to participate in the Interoperability Showcase for the first time.

We presented a demonstration of an interoperable system addressing the preparation of one care environment (the ICU) to receive a patient from another care environment (the OR). We included medical device data not currently used for preparation and showed how transferring medical device settings, using CONNECT as an information broker, can help deliver rich contextual information both directly to the caregiver preparing the environment and to decision support algorithms aiding that caregiver. CONNECT[1] is an open source software solution that supports standards-based health information exchange both locally and at the national level.

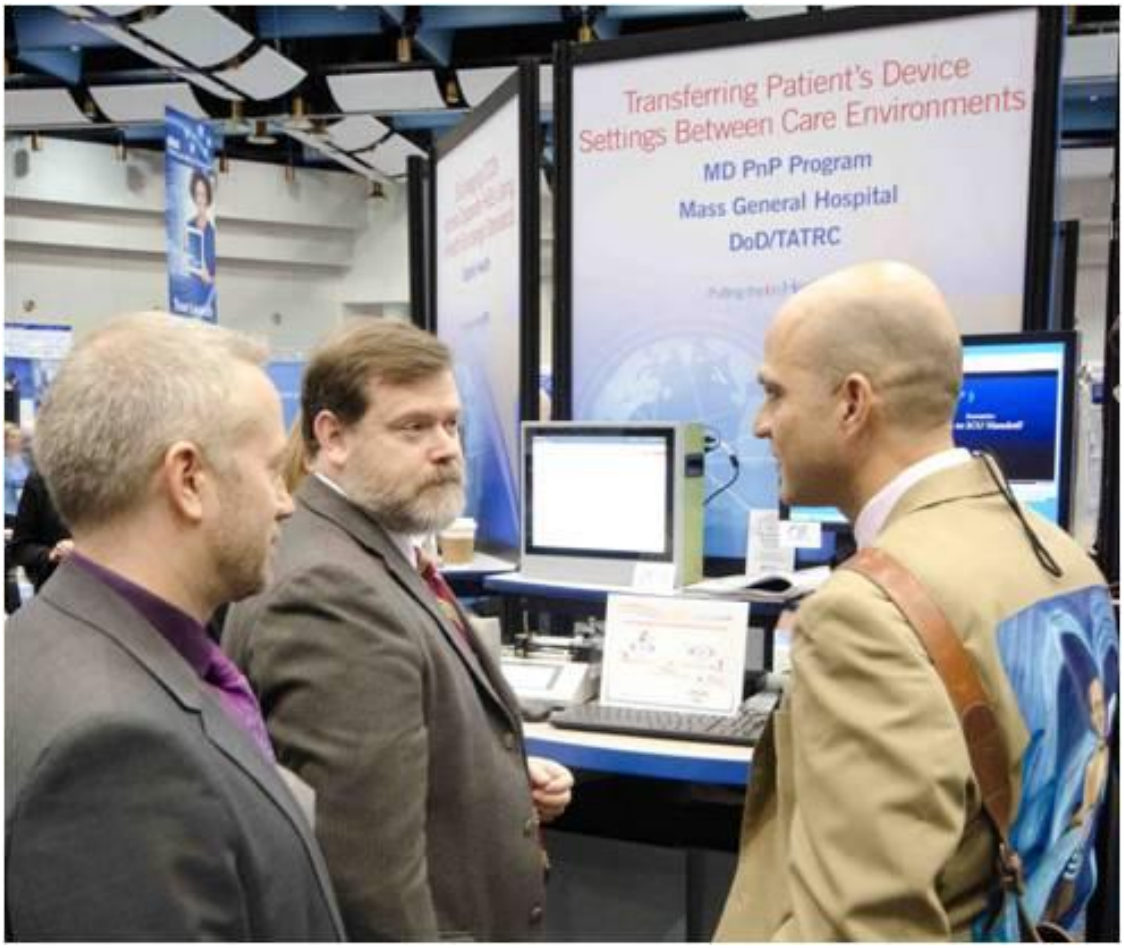


Figure 2: Dr. Julian Goldman, Principal Investigator, shows our demonstration to Dr. Farzad Mostashari, National Coordinator for Health Information Technology, and Dr. Doug Fridsma at HIMSS13

The MD PnP kiosk was selected for presentation to Dr. Farzad Mostashari, the National Coordinator for Health Information Technology. Figure 2 shows QMDI principal investigator Dr. Julian Goldman presenting the work to Dr. Mostashari and Dr. Doug Fridsma, Chief Science Officer and Director of the Office of Science and Technology for the ONC.

[1] http://www.connectopensource.org/about/what-is-connect

Overall Set-up

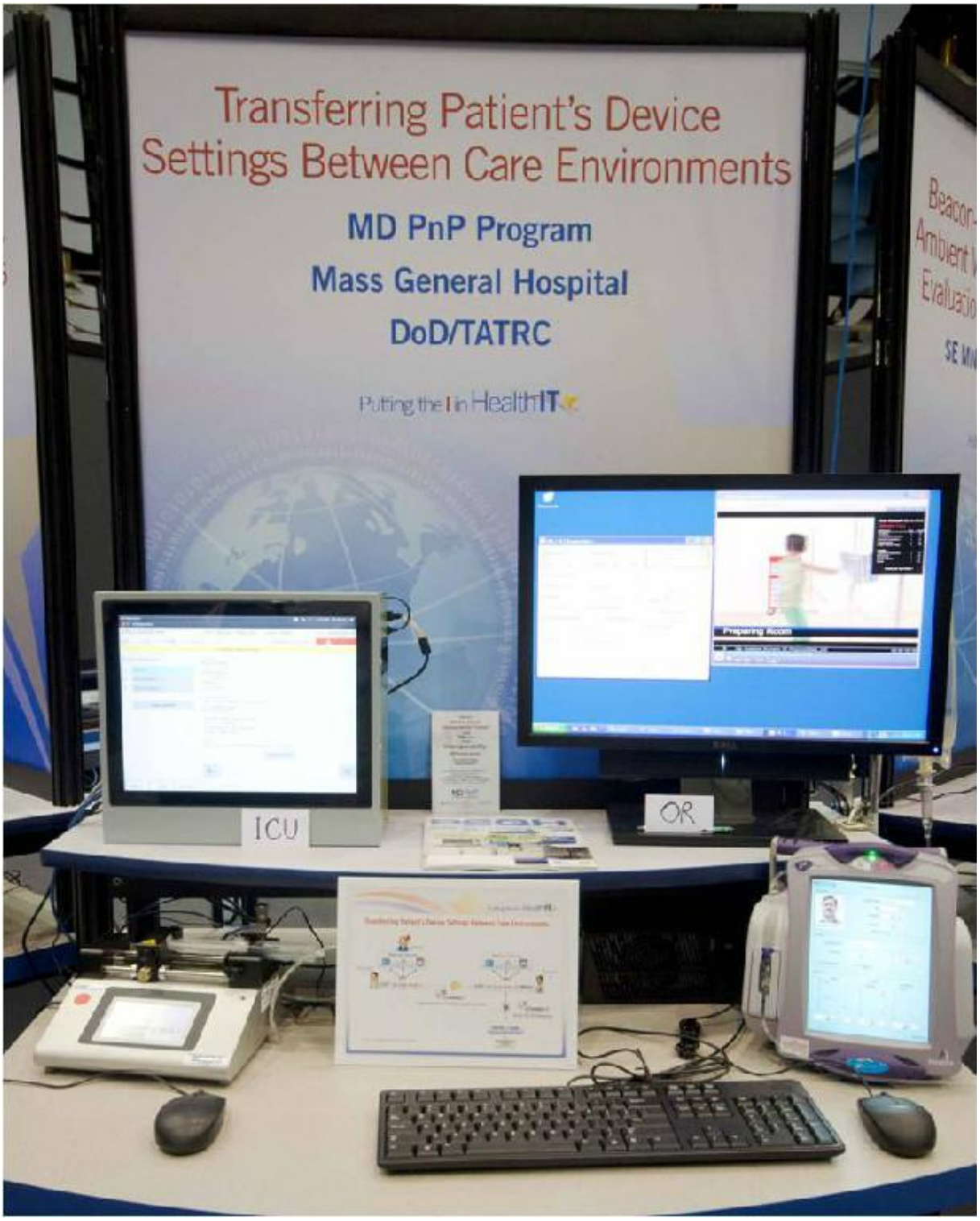


Figure 3: MD PnP demo configuration, with DocBox prototype on left representing the Intensive Care Unit and Medical Device Coordinating Framework on right representing the Operating Room.

Figure 3 shows our demo set-up at HIMSS. On the left, the Intensive Care Unit (ICU) is represented by a prototype Integrated Clinical Environment (ICE) developed by DocBox, Inc, with a programmable KD

Scientific syringe pump attached. On the right, the Medical Device Coordinating Framework (MDCF) developed by Kansas State University runs on a PC and represents the OR (Operating Room); attached is a customized version of a Hospira large volume infusion pump. A video animation of the scenario produced by DocBox runs on a display in the background.

The interface in the OR environment collects device settings from the Hospira pump as well as from a software simulated patient monitor. It also allows a clinician to enter more detailed information to accompany device settings and forwards that information, via CONNECT, to the ICU environment. The interface in the ICU receives device settings and clinician-entered information from the OR and uses that information to drive intuitive decision support for the caregiver preparing the ICU to receive a patient. In the ICU environment, patient history is also retrieved via the National Health Information Network to further aid in decision support. In our demo, the patient history is available thanks to work done by TATRC to provide a CONNECT gateway into the Department of Defense AHLTA records.

Set-up Details

An overview of the flow of the scenario is presented in Figure 4. In this demonstration, for the "ICE in Care Area 1," we used the MDCF in the OR environment. The connected medical devices are the Hospira large volume infusion pump and a software simulated patient monitor. For the "ICE in Care Area 2," we used the DocBox ICE in the ICU environment. The connected medical device is the programmable KD Scientific syringe pump. An aspect of the demo that ONC was particularly interested in was that these two care areas could, in theory, be at different geographic locations, and the data flow would work in the same way.

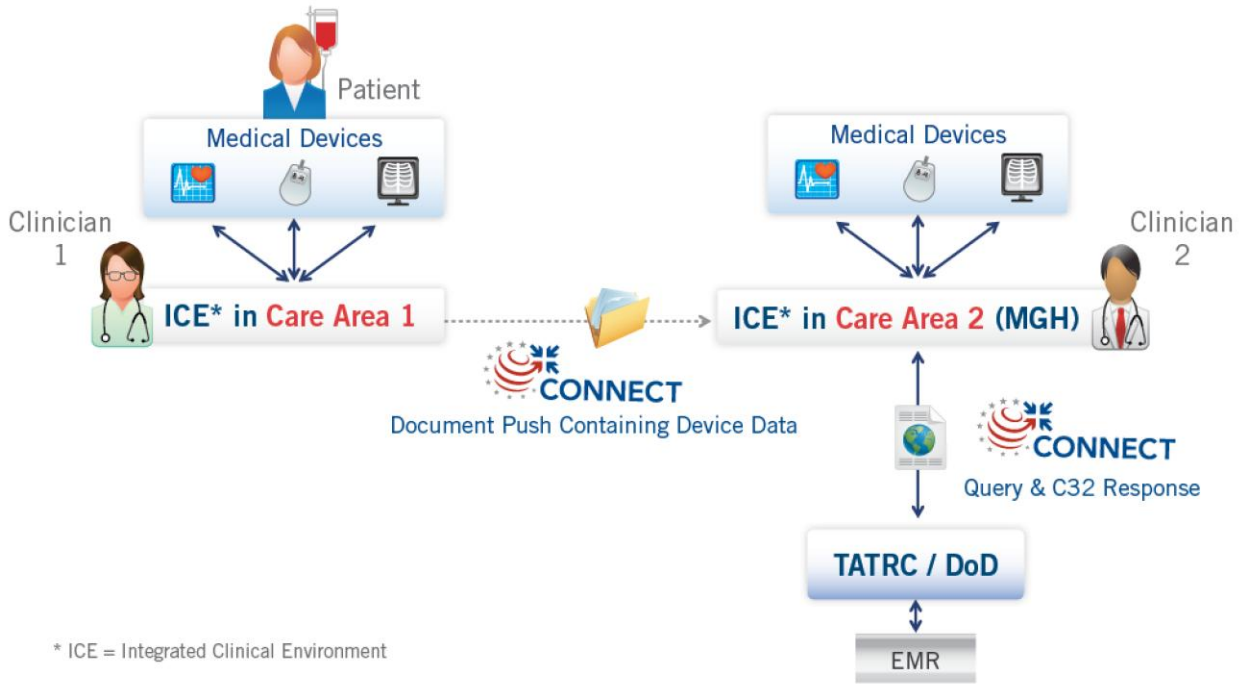


Figure 4: Overview of the Demonstrated Scenario, Connections, and Flow

Figure 5 depicts the selection of the settings transfer application and its binding to both the infusion pump device and the simulated patient monitor device.

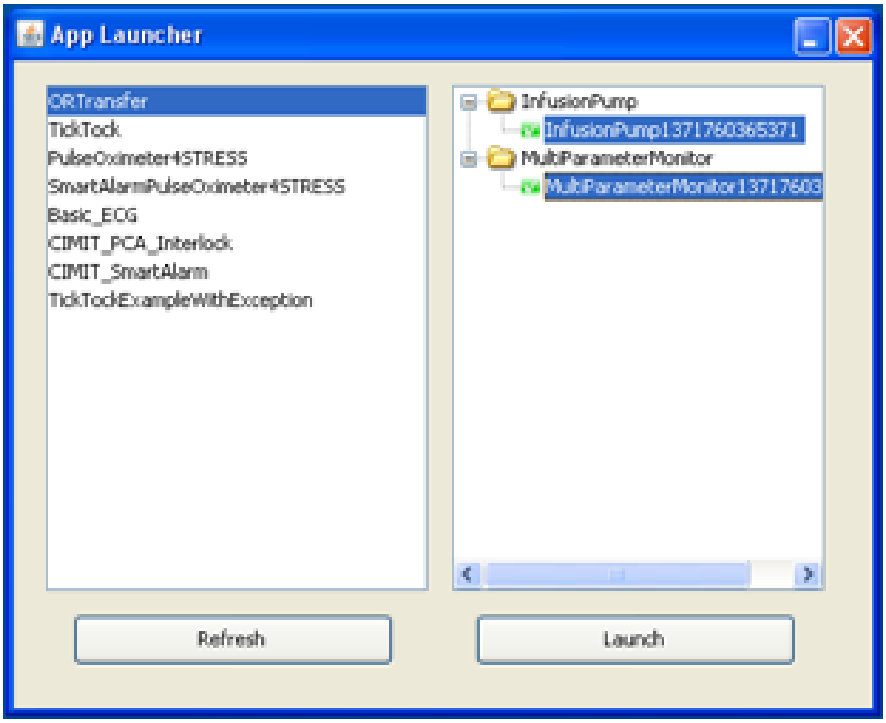


Figure 5: MDCF App Launcher

Figure 6 shows the simulated patient monitor running as a software simulation on the same PC.

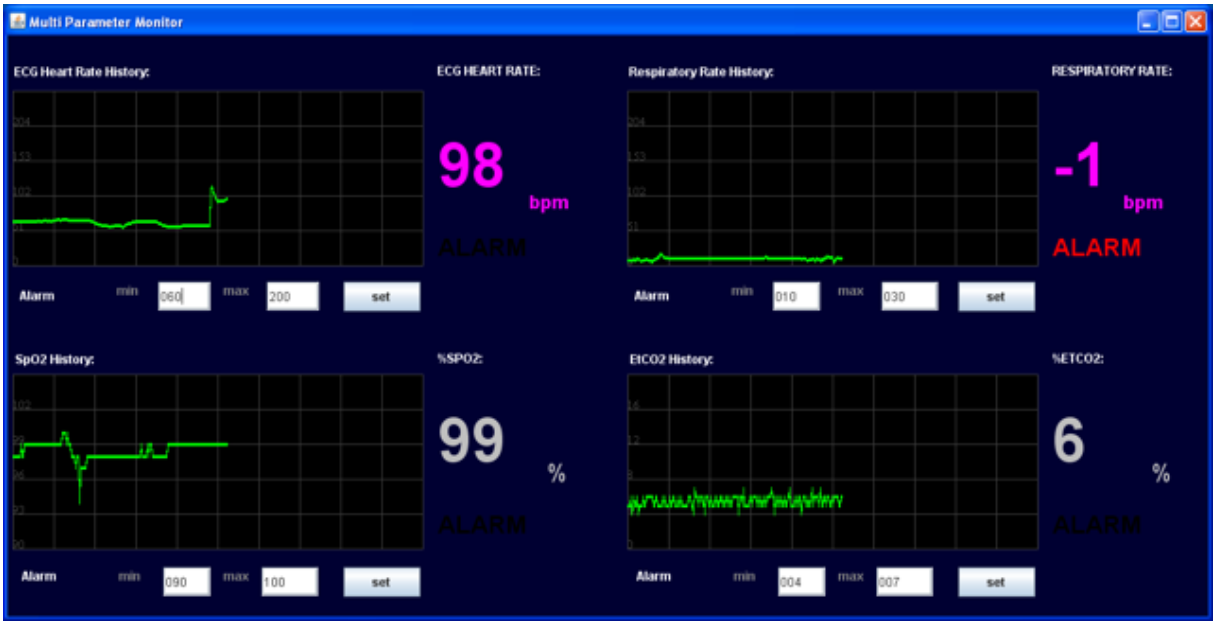


Figure 6: Simulated Patient Monitor

Figure 7 shows the user interface developed to run on our specialized Hospira pump and to connect to the MDCF and emit settings.

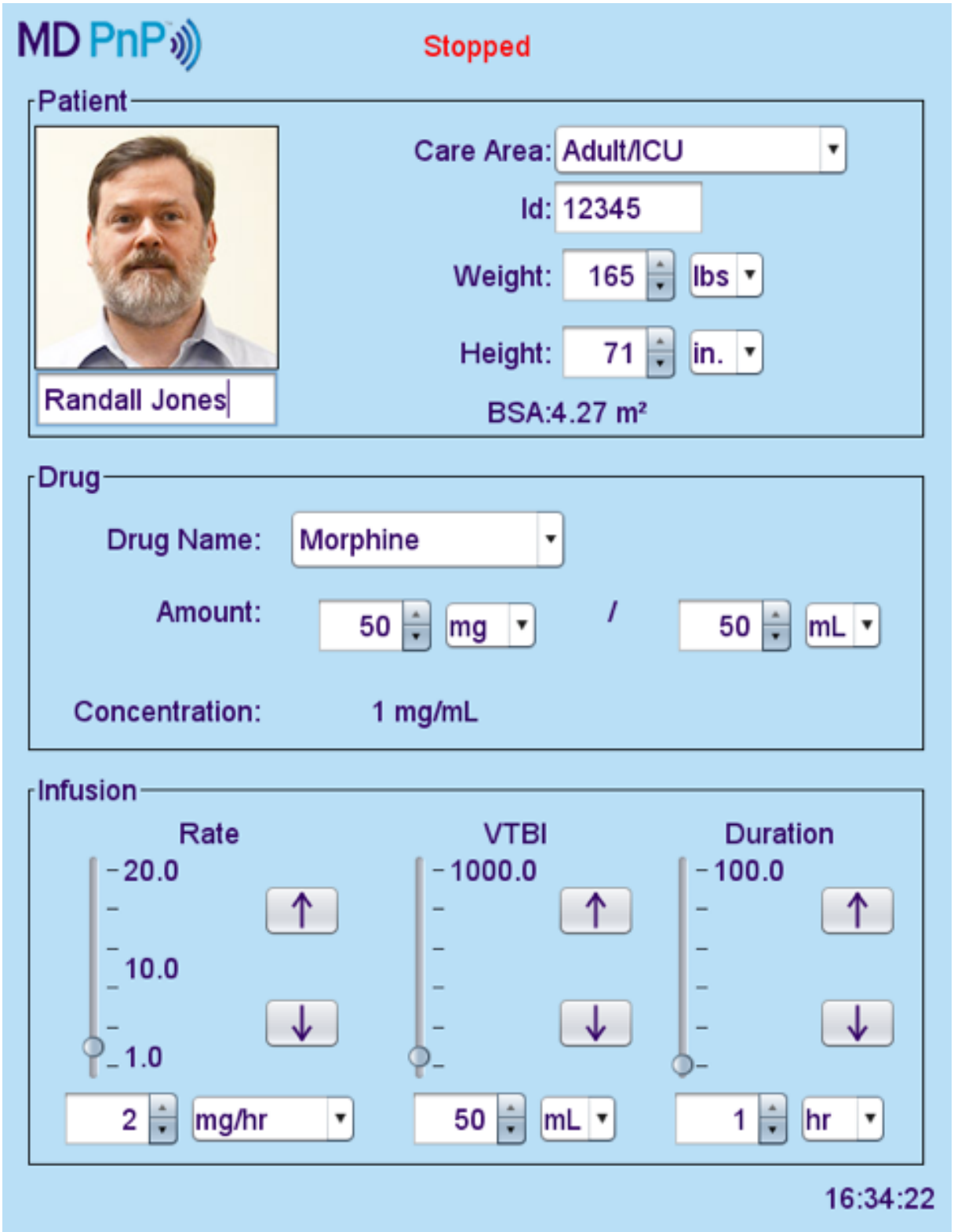


Figure 7: Demonstration Infusion Pump User Interface

Demonstration Scenario

Figure 8 shows the starting point of the scenario. The application in the OR shows that settings are available from both a Multi-parameter Monitor and an Infusion Pump. It also reflects limited patient demographics and allows a clinician in the OR to enter additional contextual information that may be relevant, such as the number of pressure channels in use and a summary of care given in the OR.

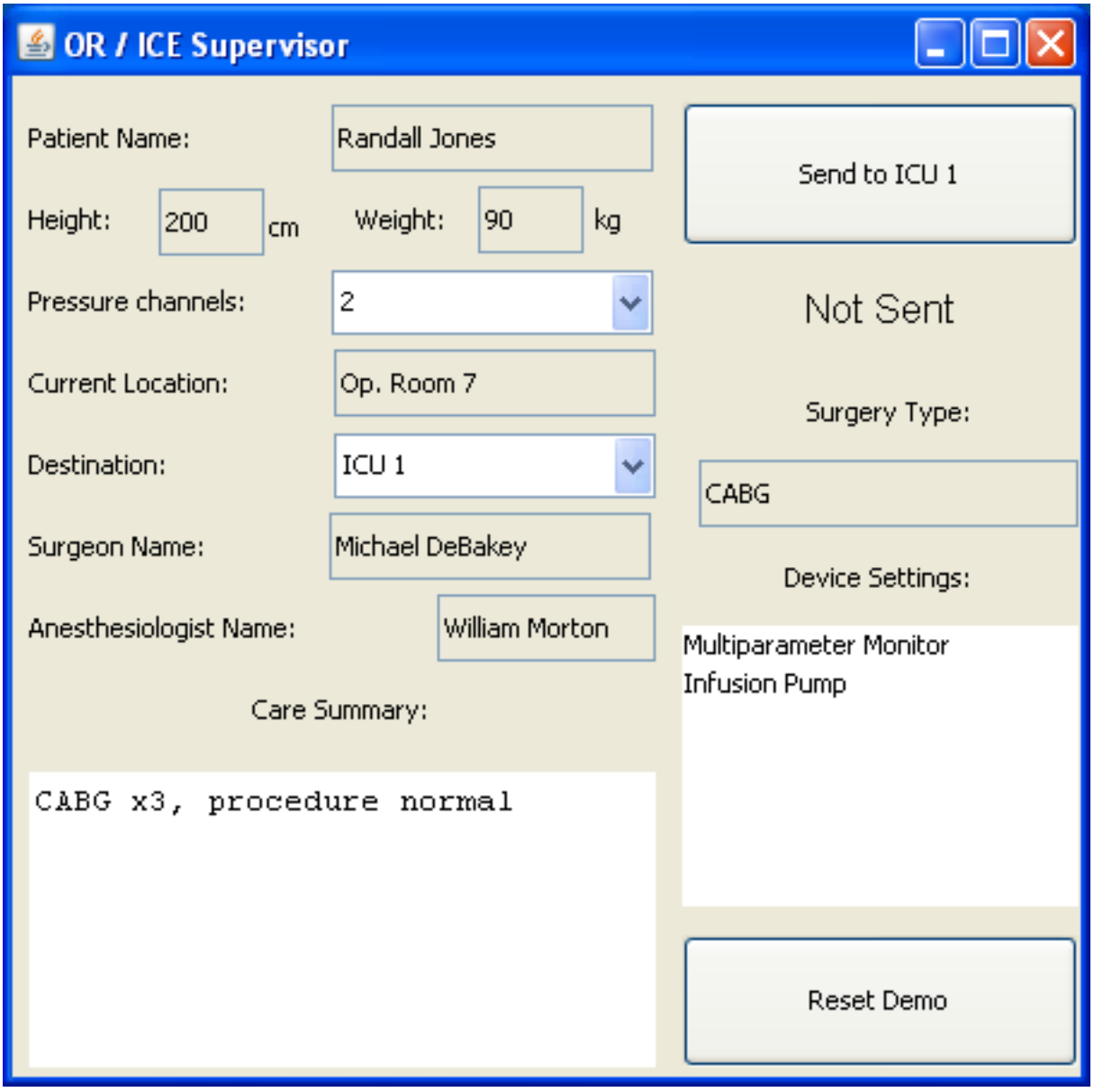


Figure 8: User interface in the OR is an application running on the Medical Device Coordinating Framework

Finally, the clinician may select the destination for the patient and transmit information to the ICU to aid in preparation. The demo then moves to the DocBox interface, where a decision support algorithm consumes the information from the OR and produces workflow aids. For example, a dynamic equipment list reflects changes to the required equipment based on the pressure channels specified by the clinician in the OR. The DocBox system also queries the National Health Information Network to retrieve historical care documentation. In our scenario, that care history includes a cited allergy to latex; the equipment list in the ICU is dynamically updated to specify nitrile gloves be prepared.

The ICU support system also prepares the infusion pump with proper programming, providing existing pump settings from the OR as well as the physician's current orders as points of reference to ensure proper pump set-up. When the assisted workflow in the ICU is complete, the clinician in the ICU selects an option to signal the OR that room preparation is complete. In this way, caregivers in the OR are provided with advance notification of readiness before they begin the patient transfer.

Audience

Most visitors to our HIMSS kiosk stopped by for only a few minutes, but this was long enough to show them the process of collecting data in the Operating Room, transmitting that data to the Intensive Care Unit for context-aware workflow, and sending an acknowledgment back to the OR indicating that the ICU has been prepared. Many visitors also watched the video animation demonstrating this scenario playing out in a virtual hospital environment and highlighting some potential problems ameliorated by the technology solution in the proposed state of the scenario that we were demonstrating.

Participants

The demo was developed by teams at DocBox Inc, Kansas State University, and the MD PnP program. It required collaboration both within and amongst the participating organizations. Attending HIMSS and presenting the demo at the kiosk were Tracy Rausch of DocBox Inc, Sam Procter and Yu Jin Kim of Kansas State University, and Dave Arney and Jeff Plourde of the MD PnP program. Kiosk coverage was also provided by PI Julian Goldman, Pratyusha Mattegunta, and Sue Whitehead (all from MD PnP).

Lessons Learned

Presenting the demo at HIMSS was a valuable learning experience for both the QMDI project and for the MD PnP program as a whole. In addition to the technical lessons and refinements to requirements that emerged from the demo development process, there was useful feedback on the demo scenarios, the presentation format, and overall interoperability program.

1) This demonstration served as a great exploration of the functional component identified as the "ICE Coordinator" in the ASTM-F2761 standard for the Integrated Clinical Environment. For the first time, we brought together several ICE environments in a larger system of multiple ICEs using commonly accepted technology (CONNECT). The challenges involved taught us that in addition to creating standardized interfaces between the Integrated Care Environment and its attendant devices and applications, we must also work to define a standardized interface for coordination of ICEs through an external interface.

2) Preparation for this demonstration exposed a need for some clarification of specific points in the ICE standard. For example, the standard defines an ICE to include one and only one patient. In preparing a room before the patient's arrival, clearly the ICE must also be able to function in an anticipatory mode. In separate work, we have witnessed the difficulty vendors have had in creating a smooth preparation workflow. For example, certain current-generation equipment in our lab will cause the discharge of a patient from the OR environment if an attempt is made to associate that patient with the ICU environment, even if the attempt is made while the patient is still in the OR.

3) Interfacing several distinct implementations of ICE also helped clarify our understanding of how we expect ICE to evolve. We realized that collaborators with distinct needs and timelines would not be likely to standardize on one canonical implementation of the standard. Through our work on this demo we have learned that we can make iterative improvements to the standardized interfaces between system components without necessarily imposing a single rigid design.

4) The most common documents providers plan to exchange via the Federal Health Architecture are documents related to continuity of care. We discovered a gap in the specification of these documents as they pertain to medical devices, and most especially the settings of those devices. Our work proposes a viable pathway for including robust data sets in those commonly accepted documents. Our example prototype continuity-of-care document containing medical device data has been shared on our program's SourceForge site (http://sourceforge.net/p/mdpnp/wiki).

5) The CONNECT software solution is built upon a technology stack comprised of myriad other standards. While this is indeed a benefit, it also introduces a high barrier to entry for those who would like to participate both as transmitters and receivers of information. For example, while CONNECT uses commonly accepted SOAP web services for communication, those services depend heavily on an array of XML schemas from a number of sources. Plumbing the depths of various standards to understand how to populate CONNECT messages can be at best tedious and at worst overwhelming. We have posted example code on our SourceForge site that demonstrates how one would populate all the fields necessary to make a successful document submission to the CONNECT system. We have also shared a small example of how to extract commonly accessed fields from data received by the CONNECT infrastructure.